Discriminative Alignment Models For Statistical Machine Translation

Nadi Tomeh

UNIVERSITY OF PARIS-SUD
DOCTORAL SCHOOL OF COMPUTER SCIENCE

P H D T H E S I S

to obtain the title of

Ph.D. of Science

of the University of Paris-Sud
Discipline : Computer Science

Defended by
Nadi Tomeh

Discriminative Alignment Models For Statistical Machine Translation

Thesis advisor: François Yvon
Thesis co-advisor: Alexandre Allauzen

prepared at LIMSI-CNRS, TLP Team
defended on June 27, 2012

Jury:

President:	Anne Vilnat	-	Université Paris-Sud
Reviewers:	Eric Gaussier	-	Université Joseph Fourier
	Philippe Langlais	-	Université de Montréal
Examinator:	Hermann NEY	-	RWTH Aachen
Invited:	Nasredine Semmar	-	Commissariat à l'Energie Atomique
Advisor:	François Yvon	-	Université Paris-Sud
Co-advisor:	Alexandre Allauzen	-	Université Paris-Sud

Remerciements

A l'issue de ces quatre dernières années de travail, je souhaiterais remercier toutes les personnes qui m'ont permis, de près ou de loin, de mener à bien ce travail.

Ma gratitude va tout d'abord à mon directeur de thèse, Monsieur François Yvon, Professeur de l'Université Paris-Sud. Je le remercie de m'avoir confié ce travail de recherche et de m'avoir guidé, encouragé, conseillé, tout en me laissant une grande liberté de recherche tout au long de cette thèse. Je tiens également à remercier Monsieur Alexandre Allauzen, Maître de Conférence de l'Université Paris-Sud et co-encadrant de cette thèse, pour sa disponibilité, ses idées et son aide au quotidien. Au-delà de leur rigueur scientifique, je voudrais saluer du fond du coeur leurs qualités humaines qui ont facilité mon intégration au sein du LIMSI et en France.

Mes remerciements s'adressent également à Madame Anne Vilnat, Professeur de l'Université Paris-Sud, pour m'avoir fait l'honneur de présider mon jury de thèse. Je suis aussi très reconnaissant à Monsieur Eric Gaussier, Professeur de l'Université Joseph Fourier, et Monsieur Philippe Langlais, Professeur de l'Université de Montréal, d'avoir accepté de juger ce travail et d'en être les rapporteurs. Enfin, j'aimerais remercier les autres membres du jury, Monsieur Hermann Ney, Professeur de l'Université RWTH Aachen, et Monsieur Nasredine Semmar, Chargé de Recherche au CEA, pour leurs questions, remarques et suggestions.

Ce travail a été effectué au Laboratoire d'informatique pour la Mécanique et les Sciences de l'ingénieur (LIMSI-CNRS) dont je remercie le directeur, Monsieur Patrick Le Quéré, de m'avoir ouvert les portes. Je remercie également Monsieur Jean-Luc Gauvain, Directeur de Recherche CNRS, de m'avoir accueilli au sein de son équipe, Traitement du Langage Parlé (TLP).

J'ai eu la chance au cours de ma thèse de partager le bureau de Guillaume Wisniewski, qui est devenu plus qu'un collègue pour moi. A son contact, et grâce à ses nombreuses qualités (y compris celle de DJ), je pense avoir énormément appris, pas seulement sur l'apprentissage statistique, mais aussi sur la vie. Grâce à lui, venir au bureau tous les matins (ou presque) était un grand plaisir, merci Wichnou.

J'aimerais particulièrement remercier Aurélien Max pour son accueil chaleureux dès mon premier jour au LIMSI, pour sa gentillesse, son soutien continu, son intérêt dans mon travail, ainsi que pour les nombreuses discussions enrichissantes que nous avons eues.

Je tiens à remercier infiniment tous mes collègues enseignants à l'Institut Universitaire de Technologie (IUT) d'Orsay pour leur confiance pendant mes années de monitorat et d'ATER. Mes remerciement vont notamment à ma tutrice Cécile Balkanski pour son aide précieux au début du chemin, et également à Hélène Bonneau-Maynard pour ces compliments qui m'ont toujours fait plaisir.

Grâce à l'ensemble de mes collègues de travail au LIMSI, j'ai pu travailler dans un cadre exceptionnellement agréable. Je pense particulièrement à Thiago qui m'a appris comment vivre sans trop s'inquiéter, Ilya et Nadège sur qui j'ai toujours pu compter, Thomas pour son suivi méticuleux de l'état d'avancement de ma thèse, Penny pour les promenades au soleil, Houda la spécialiste de l'administration française et l'organisatrice de soutenances, et à Artem, Cécile, Clément, Eric, HaiSon, Hervé, Marianna, Nicolas et Souhir. Merci à tous

pour votre bonne humeur, pour nos séances de rires et pour nos discussions autour de cafés, beaucoup de cafés.

Ces remerciements ne seraient pas complets sans une pensée pour mes amis qui n'ont cessé de me rappeler qu'il existait un monde à l'extérieur du LIMSI. Je pense notamment à Lynn, ma meilleure amie, et à son prince charmant Nicolas. Un grand merci également à Charlotte et Tony pour m'avoir changé les idées quand il le fallait, et à Christine pour m'avoir appris mes premiers mots d'allemand. Mes pensées les plus chaleureuses vont par ailleurs à mes meilleurs amis de Damas, particulièrement à Dani, Louay, Mike, Micho, Tony et enfin Lana pour leurs encouragements et les fous rires que nous continuons à avoir malgré la distance et les circonstances actuelles. J'aimerais enfin remercier mon camarade Jean-Baptiste pour son soutien dans toutes les batailles que nous avons menées ensemble.

Je voudrais exprimer ma profonde reconnaissance à mes parents Nayla et Nazih et à ma petite soeur Ansa, qui m'ont encouragé tout au long de mes études avec tendresse et malgré toutes les difficultés qu'entraîne la distance. J'espère qu'il sont fiers de moi.

Pour finir, j'aimerais remercier la jeune, belle, charmante et intelligente Anne-Sophie qui a toujours su me soutenir. C'est à elle que je dédie ce travail.

Contents

Introduction

Natural language translation is the communication of the meaning of a text in the source language by means of an equivalent text in the target language. A collection of such source texts along with their translation is referred to as a *bitext* or a *parallel corpus*. The rich linguistic knowledge embedded in a bitext is valuable for many practical applications in natural language processing, especially in the era of the Internet, when the body of multilingual communications and translated texts is growing at a fast pace. While the most predominant application for bitexts is Statistical Machine Translation (SMT), they are used in multilingual (and monolingual) lexicography, word sense disambiguation, terminology extraction, computer-aided language learning and translation studies, to name a few. The potential of such translated resources is amplified when the hidden translation relation is revealed, and the correspondence between text units is found. Although each text is typically represented in a plain form, it possesses an intrinsic hierarchical structure composed of letters, words, phrases, sentences, paragraphs, etc. While the translation process establishes an equivalence relation between whole structures, bitext alignment aims to explicit this relation between smaller text units at various levels of granularity.

The task of building bitexts and aligning them is well-defined only in the context of a given application. For instance, translation studies or bilingual reading applications require clean bitexts with known translation directions and other meta information. Alignments must also be very accurate and cover all the words in the bitext. However, data-driven applications such as SMT and multilingual information retrieval seek the regularities in large amounts of parallel data, with focus on frequent words and expressions. Therefore, the noise in the bitext and its alignment can be easily dealt with. State of the art SMT systems, including phrase-based (Koehn, Och, and Marcu, 2003) hierarchical-based (Chiang, 2005), and syntax-based (Galley et al., 2004; Melamed, 2004), learn translation rules from large corpora of parallel sentences in two steps. First, the parallel sentences are aligned at the sub-sentential level; translation rules are then extracted and evaluated to build the translation model. Put informally, learning statistical translation models is made possible by the knowledge of alignments, since they provide the necessary annotation from which translation decisions can be learned. Figure 1 shows an example of an Arabic-English parallel sentence and its alignment. In this dissertation, we will consider the task of automatically obtaining such

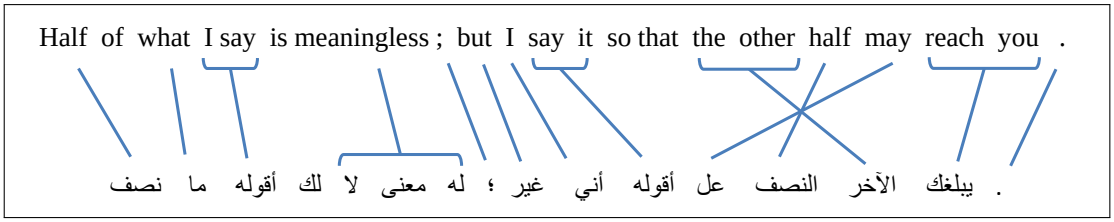


Figure 1: *A sub-sentential level alignment for an aphorism from "Sand and Foam" by Khalil Gibran (1974).*

alignments for arbitrary parallel sentences. While our focus is on improving the translation quality of a phrase-based system, the improvements obtained in alignment quality would likely benefit the other applications as well.

Bitext alignment is an arduous task because meaning is not expressed seemingly across languages. It varies along linguistic properties and cultural backgrounds of different languages, and also depends on the translation strategy that have been used to produce the bitext. Therefore, the corresponding units do not necessarily have the same structural role and position in their own languages. Indeed, a word in a language often matches a morpheme in another, or contrarily, a whole phrase. This is all the more so true for non-literal translation styles.

Current practices in bitext alignment

Recent advances in the field of machine learning, accompanied by increased computational power have contributed to the development of data-driven, statistical approaches to solve the alignment problem efficiently. At the center of such approaches are statistical models, learned from the data, and used to evaluate and to select among alternative alignments the "best" fit for a given bitext. In order to reduce the complexity, alignments at different levels of granularity are produced separately: in a collection of bitexts, documents are first aligned, then sentences inside them and finally words and phrases within the parallel sentences. The focus of this dissertation is the sub-sentential alignments.

The early approaches modeled the alignment as a hidden variable in the translation process (Brown et al., 1993). These models produce asymmetric, one-to-many alignments, because each target word is assumed to be generated from one source word. However, this assumption is over-simplistic since word alignments are in general symmetric and many-to-many. The current practice to achieve symmetry is to build two one-to-many alignments in opposite directions, and combine them using a symmetrization heuristic (Och, Tillmann, and Ney, 1999; Koehn, Och, and Marcu, 2003; Och and Ney, 2003). These word alignments are used in many applications, including modern phrase-based SMT systems. Typically, word alignments are computed for large amounts of parallel sentences, then for each of which, a heuristic is used to extract phrase pairs that are consistent with the word alignment (Koehn, Och, and Marcu, 2003). The extracted phrase pairs are used then to train the translation model.

Issues and challenges

The alignment problem is not solved and is currently an active research area. Generative approaches are widely used in practice. They require a large amount of data to deliver a good performance, and their computational complexity is one of their major issues. They model one of the parallel sentences, or both of them, in addition to the alignment variable. This results in an overhead complexity, which requires strong independence assumptions in order to cope with. Moreover, incorporating features is prohibitively expensive. The alignment model must take into consideration the alignment structure and the interaction between alignment decisions. However, modeling these dependencies adds to the computational complexity which can rapidly become prohibitive. Compared to alignment produced by human experts, state of the art alignment systems produce many errors. For example, on an Arabic-English parallel corpus, IBM model 4 (Brown et al., 1993) produced a 23% error rate, measured by Alignment Error Rate (AER), which is a combination of precision and recall on word links[1]. It is important to improve the alignment quality since it should improve the translation quality:

[1]The details of these experiments are given in Section 4.9.10

less precision errors enhances the quality of extracted translation rules; and less recall errors increases the number of such rules.

It is not clear, however, how improvements in alignment quality measured at the level of words, are reflected in phrase-based translation systems. This is mainly due to the interaction between the alignment and the translation rule extraction step. The "standard" extraction heuristic tends to neutralize some improvements and propagate some errors. Furthermore, it does not enable any control over the number of extracted phrase pairs, and does not take the difference in their quality into consideration. For example, the phrase pairs extracted from the English-French Europarl corpus (Koehn, 2005), contains 467 distinct translations for the phrase "`European commission`", and 672 distinct translations for "`!`". Many of these translations are inaccurate, however, the heuristic does not provide any mechanism to differentiate between them during extraction. The majority of extraction methods are based on the links in the one best word alignment whereas the other good alignments according to the model are ignored. This is mainly because computing link posterior probabilities to evaluate individual links under the entire alignment model is intractable for complicated models. Furthermore, phrase extraction procedures typically rely only on word or phrase alignment models, which are error-prone and are trained using objective functions that correlate only indirectly with the translation task.

Improving alignments with discriminative techniques

In this dissertation we address the problems of word alignment and phrase pairs extraction. We improve the state of the art in several ways using discriminative learning techniques. Empirical results show significant improvements in the alignment quality as measured by AER and the translation quality as measured by Bilingual Evaluation Understudy (BLEU).

Our first objective is to improve the intrinsic alignment quality and see how the improvements correlate with the translation quality. We present a discriminative word alignment model which recast the problem in a symmetric way. The alignment matrix is modeled directly, requiring that an alignment decision must be made for every word pair. The discriminative framework enables to model only the alignment variable, which matches the actual use of the model in alignment prediction. Furthermore, incorporating additional features is less expensive than for generative models. Discriminative models typically requires training data annotated with alignment information, which can be helpful to learn what form alignments are expected to have. Unlike generative models, a relatively small amount of annotated data is sufficient to achieve state of the art performance. In order to take the interaction between links into consideration, we find a middle-ground solution, using machine learning stacking techniques, to model the structure indirectly without additional complexity. This framework leads to significant improvements in alignment quality as measured by the AER on an Arabic-English corpus, which carry on to the translation quality when using the standard phrase pairs extraction method for large-scale Arabic-English NIST'09 data.

Our second objective is to enhance the phrase pair extraction procedure so as to make a better use of the entire alignment distribution, and not only the "best" alignment. A key concept to the success of such a method is an accurate estimation of the link posterior probabilities. In our framework, we model these posteriors directly, in an inexpensive way. This enables to use an alternative posterior-based extraction method, which is more sensitive to improvements in alignment quality than the standard heuristic. Using the link posteriors helps controlling the error propagation from the alignment model to the translation model. Additionally, a finer control of the number of extracted phrase pairs is made possible so as to balance the phrase pair precision and recall. This is extremely helpful to adapt the extraction to the size of the available training data. Applying this method yields further improvements in translation quality.

We push this approach one step further in order to incorporate additional useful information to the extraction process. Similar to the word alignment, the reformulation of the problem in a supervised framework offers a principled way to combine several features to make the procedure more robust to alignment difficulties. However, obtaining annotations for supervised learning is the main obstacle. To overcome this obstacle, we propose a simple automatic method to label phrase pairs according to their utility to translation, which enables incorporating the translation quality into the procedure. Thus, we obtain a set of phrase pairs, labeled as useful, which we use as input to machine learning techniques that permit learning from positive data only. The outcome is a model that distinguishes useful phrase pairs from the others. This method produces enhancements in translation quality, in addition to a better exploration of the space of possible phrase pairs than the previous extraction methods. The same approach can be extended straightforwardly to other applications that use phrase alignments. The only requirement is to be able to identify examples of the desired category of phrase pairs.

These ideas, along with empirical results, are discussed in this dissertation. It is organized in two parts. Part I provides an overview of the bitext alignment problem and of its applications, with a synthetic view of existing methods from the literature. We start in Chapter 1 with a detailed overview of the alignment problem from a linguistic point of view. We point out its difficulties and present a generic framework to solve it. We also describe and compare several evaluation methods for alignment quality. In Chapter 2, we provide the reader with a detailed exposition of the state of the art alignment methods. We start by asymmetric word-based alignment methods, including unsupervised generative approaches and supervised discriminative approaches. Then, we describe symmetric word-based approaches in Section 2.3. Such methods include symmetrization methods which operate on the output of asymmetric methods, methods that use weighted matrices, and methods that score global alignment structures. In Section 2.4, we present a different alignment paradigm based on synchronous grammar and discuss the role of syntax in alignment. Phrase-based alignment models are described in Section 2.5. This includes bisegmentation models and generalized model to which the various extraction methods belong. In Section 2.6 we describe several cues and correlation that may help the alignment algorithm and explain how they can represented as features. Part I ends with Chapter 3, which describes the various components of a state of the art phrase-based translation system. It also provides details on how translation models are typically built from alignments.

Part II presents our original contribution concerning the use of discriminative learning techniques to improve the alignment quality for statistical machine translation. We present our discriminative word-based alignment framework in Chapter 4, and provide empirical evidence of the improvement in alignment quality as measured by AER. In Chapter 5 we study the impact of our alignment models on translation performance when using the standard extraction heuristic. We also present an alternative extraction procedure that benefits from the capability of our model to provide accurate estimates of link posteriors. We provide a set of experiments showing improvements in translation quality as measured by BLEU using both methods. In Chapter 6, we reformulate the general phrase alignment problem in the supervised discriminative framework and show how single class classification techniques can be used to solve it. The correlated experiments show improvements in translation quality.

Part I

Bitext Alignment

CHAPTER 1

The Alignment Problem: An Overview

Translation alignment aims to reveal the hidden structure of translation and to establish correspondences between the elements of the translated texts. Such insight to the translation process allows for a better understanding and exploitation of translated texts in numerous applications. Translation involves the transfer of a text from one language into another while conserving its meaning. The alignment task is then to recover the units of text used to perform the transfer. Issues arise from the non-deterministic nature of translation which results in many ways to re-express the meaning using translation units of variable granularity, ranging between individual words and entire texts. While the granularity of translation units in a text can not be determined beforehand, many applications of alignment and algorithms rely on some assumptions. One such assumption is that the alignment can be established at the level of words, which does not always match the reality.

After defining the generic bitext alignment problem in Section 1.1, we present in Section 1.2, a brief discussion regarding some of the inconsistencies between translation and alignment. In Section 1.3 we define the most frequently used alignment assumptions corresponding to different level of the text hierarchy including documents, sentence and words. Applications of these alignment tasks are then discussed in Section 1.4. Sections 1.5 and 1.6 introduce the generic computational framework in which we propose to consider the alignment problem. They also provide the reader with an overview of the alignment search space constraints and the correlations and cues used to guide the search for the best alignment. Finally, several intrinsic and extrinsic evaluation methods of alignment quality are presented in Section 1.7.

1.1 Bitext Alignment

Translation is the process of transferring a text from a source language to a target language. Each text is related to a specific socio-cultural context in which it should function properly. Translation is described in terms of literal rendering of meaning, adherence to form. The process of translation establishes a relation between a text[1] in the source language and its translation in the target language. The source and target texts placed alongside each other are called a *parallel text* or a *bitext*.

[1] A text can be of any length: a document or a sentence or a collection thereof.

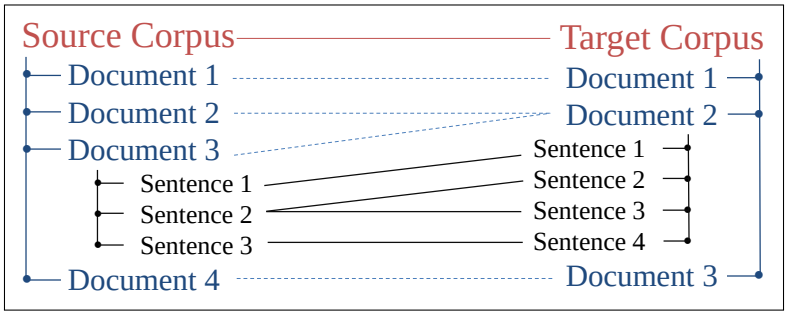


Figure 1.1: *Hierarchical structure of a bitext and alignments at each level in the hierarchy.*

The task of *bitext alignment* is the identification and linking of corresponding linguistic units in both halves of the parallel text[2]. The text on each side of a bitext decomposes hierarchically into documents, which similarly decompose into sentences and then words[3]. The alignment aims, then, to explain the coarse translation equivalence relation, established at the root level of the bitext structure, in terms of finer units at different levels of granularity, i. e. documents, sentences and words. Figure 1.1 depicts the hierarchical structures in a bitext and alignments at the document and sentence levels. Figure 1 describes a word alignment.

1.2 Translation and Alignment

The presence of an alignment at some granularity and the difficulty of obtaining it is characterized by the so called *translation unit*. The term translation unit refers to "the linguistic level at which the source text is re-codified in target language " (Shuttleworth and Cowie, 1997). In other words, the element used by the translator when working on the source text and the carrier of the atomic unit of meaning.

In this section, we describe the interaction between the translation unit and the alignment. Particularly, we show that the translation unit can correspond to any level of the text structure (documents, sentences or words) depending on many factors, including the type of texts to be translated, the purpose of the translation, etc. We argue that obtaining alignments at a finer granularity than the translation unit is difficult; and that the obtained alignments are strongly context-dependent, and difficult to be reused in other contexts.

1.2.1 Identifying the Translation Unit

Translation units do not always correspond to words for two main reasons: the meaning-language interface; and the translation strategy. The sentence is usually assumed to be the "natural" equivalence unit: it is self-contained and meaningful grammatical structure, which would not normally be divided during translation (Newmark, 1988).

1.2.1.1 Meaning-language interface

Words and concepts Meaning does not always factor into single words, the word is clearly not the only unit of translation. From a linguistic point of view, Vinay and Darbelnet (1958) reject the word as a unit of translation since translators focus on the semantic context rather than on the formal properties of the individual words. For them, the unit is "the smallest segment of the utterance whose signs are linked in such a way that they should not be translated individually". Different languages have different *compounding*, *agglutinativity* and *morphological* characteristics, which means that expressing the same concept require a variable number of word tokens. Illustrative examples of such non-correspondence at the word level

[2]The term "alignment" is a misnomer. In computer science, it technically implies that aligned units are paired one-to-one and occur in the same order in both objects. This implication does not hold in translation. Translation does not preserve word order neither one-to-one relation. Nevertheless the term alignment has continued to be used for word pairing.

[3]The detailed hierarchy of a text contains other elements such as paragraphs, phrases, clauses, etc.

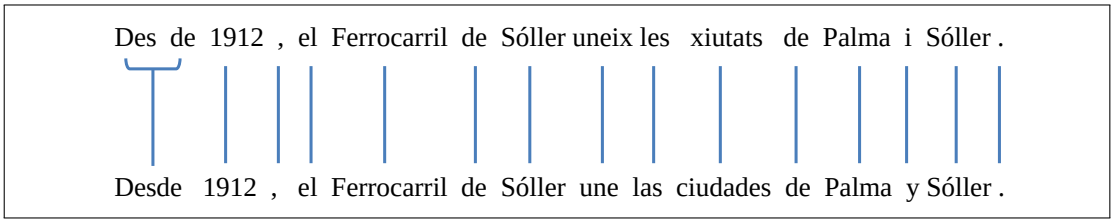


Figure 1.2: *An example of Catalan to Spanish literal translation. Translation is monotonic and word-to-word.*

abound in translations, for example the French "tout de suite" is translated to English as "immediately". The problem is worse for distant language pairs, an example is the classic Arabic word token أَنلزمكموهَا (AnulzimukumouhA) which translates into:

Persian as: "آيَا ميتوَان شمَا رَا بر آن وَادَار كنيم در حَالي كه از آن كرَاهت دَاريد",
French as: "devrions-nous vous l'imposer",
English as: "shall we constrain you to (accept) it", and
Turkish as: "onu size zorla mi kabul ettireceğiz".

Word lexical ambiguity Without the context, word-to-word matching is under-determined. Homonyms, such as the pair *left* (past tense of leave) and *left* (opposite of right), and polysemes, such as the pair *book* (a collection of pages) and *book* (to register), are two clear situations where information outside the word itself is needed to prefer one meaning over the others.

> The simplest message conveyed by the means of natural language has to be interpreted because all the words are polysemic and take their actual meaning from the connection with a given context and a given audience against the background of a given situation (Ricoeur and Thompson, 1981).

Word order An additional major obstacle is that words representing the same concepts do not occur in the same positions in input and output sentences. English for example, or romance languages such as French, are said to have an *SVO* structure since typical sentence order is *(subject-verb-object)*; Japanese and Turkish are largely *SOV*; while classic Arabic is mainly *VSO* but admits a *free-word-order* scheme similar to Latin, where the order of constituents is not strictly regulated. Consequently, the alignment model has to explicitly adopt a mechanism for *reordering* translated words into their final positions in the output.

1.2.1.2 Translation strategy

In addition to the meaning-language interface, the size of the translation unit is directly related to the translation strategy being *literal (formal)* or *free (dynamic)* (Hatim and Munday, 2005).

On the one hand, *literal (formal)* translation preserves the form with translation units being very much centered on adherence to the individual word, which makes the translation units fine-grained and easy to spot. Such formal equivalence attempts to render the text word-for-word, at the expense of natural expression in the target language. Figure 1.2 shows an example of a literal translation. Meaning is packaged and transferred in small units. [4]

On the other hand, *free (dynamic)* translation aims at capturing the sense of a longer stretch of language, resulting in meaning packaged in longer coarse-grained, hard-to-identify

[4]Notice that such translations are not so common when the languages in question are more distant, and are usually of a poor quality.

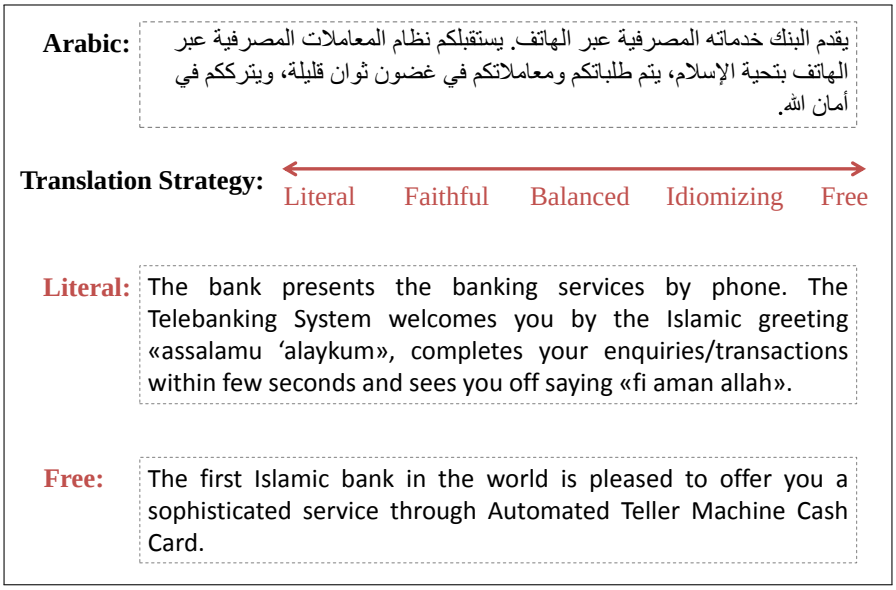


Figure 1.3: *An example of literal and free English translation of an Arabic advertisement. A translation strategy closer to literal is more faithful to the form of the original text than to its sense.*

translation units, usually at the level of phrases and clauses. Such equivalence is called "'functional"' since it attempts to convey the thought expressed in the source text, at the expense of literalness, original word order, the source text's grammatical voice, etc. Figure 1.3 develops an example from an advertisement promoting cash dispensing services given by Hatim and Munday (2005) to illustrate the difference between free and literal translations and the alignments they imply. The literal translation does not function in a population with little or no Arabic skills to appreciate the nuance. Therefore, the alternative translation had greatly departed from the form of the source text to convey its semantic and take its type into account, its purpose and its targeted audience and their socio-cultural values. Under a dynamic translation strategy, translation units can go beyond words, collations and idioms, such as thematic and information structure, cohesion and pragmatics.

The choice of a translation strategy depends on the focus and the purpose of the translation. Whether it is the form of the source text or its content (or both); whether it is the target text form and content or its reader, etc. It also depends on the translator and his preferences, interests and ideology.

The translation strategy also depends heavily on the text type. A legal text might require a much closer, more literal translation than a piece of poetry. While the sense can always be translated (Jakobson, 1959), the form often can not, due to linguistic divergences between languages including grammatical and syntactic structure. The point where form begins to contribute to sense is where we approach untranslatability. This clearly is most likely to be in poetry, song, advertising, punning and so on, where sound and rhyme and double meaning are unlikely to be recreated in the target language.

The literal versus free divide does not oppose a pair of fixed opposites, but a continuum. Translations can be positioned at any point between the two ends as reproduced in Figure 1.3.

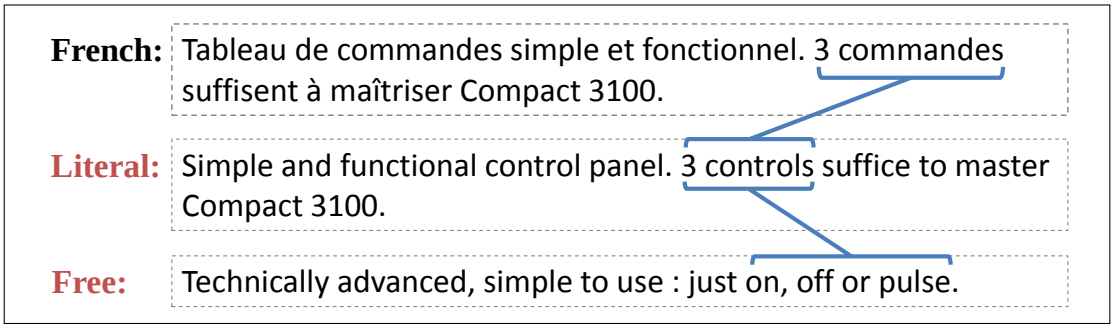


Figure 1.4: *Literal translation is explained using fine-grain translation units with direct correspondences ("3 commandes" - "3 controls"). Free translation incorporate more context making re-usability difficult in different context ("3 commandes" - "on, off or pulse").*

1.2.2 Translation Units and Alignment Difficulty

The exploration of the translation equivalence relation and the finding of the alignments is naturally done iteratively. Starting from the identification of parallel documents, down to parallel paragraphs and sentences and finally to parallel words and phrases.

Alignments are hard to identify within a translation unit because it is translated atomically. This can be clearly seen in examples. In the formal translation in Figure 1.2, the fine-grained translation units, centered around the words, allow to easily obtain a word-to-word alignment. Similarly, an easy word-to-word alignment can be obtained for the literal translation of Figure 1.3. However, for the dynamic translation where the translation unit is the sentence, sub-sentential alignments become harder to obtain, whereas alignments can always be obtained at a coarser level than the translation units.

In the majority of cases, the translation units lay somewhere between the word and the sentence, and rarely cross its boundary. Therefore, sub-sentential alignment is difficult, while sentence alignment is relatively easy; despite the fact that transpositions and rearrangements may sometimes occur.

1.2.3 Translation Unit and Alignment-Context Bound

Since translation units are usually not decomposable, alignments at a finer grained level are context-dependent. The interpretation of such alignments is left to the final application.

For example, current machine translation systems use aligned units discovered in bitexts to automatically translate a new text, and possibly in a different context. Let us consider the example in Figure 1.4. While aligned translation units: "tableau de commandes - control panel" is still valid in a different context, "3 commandes - on, off or pulse" is not. In this example, a strongly context-dependent translation is preferred for the sake of comprehensibility.

The point being made here can be further illustrated with the following example, taken from two translations of the Arabic absurdist drama by Tawfik Al-Hakeem (1960) shown in Figure 1.5. Such translations fall somewhere in a spectrum of translation approaches ranging between dynamic and formal equivalence (Nida and Taber, 2003). Now, what does it mean to

	English	Arabic
Executioner:	Now that I have warned you of this condition, do you still want me to sing?	الآن وقد لفت انتباهك إلى هذه الحالة، هل أغني؟
Condemned:	Go ahead.	غني. *(sing)*

Figure 1.5: *A sample dynamic translation from an Arabic play by Tawfik Al-Hakeem (1960).*

align "غني (Sing)" to "Go ahead" is a question that cannot be side-stepped so easily. As Martin Kay puts it in his preface to *Parallel Text Processing* (Véronis, 2000): "at the very least, it seems that it will have to mean different things to people with different purposes". As an entry in a bilingual dictionary, it might constitute a source of frustration, but for someone interested in textual pragmatics or textual salience and dynamism it might stimulate important insights.

1.3 Alignment Granularity

We briefly mentioned in Section 1.2 that equivalence can be decomposed into smaller textual units, The granularity of such units depending on linguistic properties of the text. Starting from the bitext aligned only at the root level, finer and finer alignment granularity is obtained sequentially. In this section we describe the main three levels of granularity of alignment: document level, sentence level and sub-sentential level (phrase, chunk, word, etc.).

1.3.1 Document Alignment

The first alignment problem we consider is the construction of parallel corpora by aligning documents. Building such a corpus from a multilingual data collection comprised of several documents, requires preprocessing the text into words and sentences and then performing the alignment.

Data collection of parallel documents that can be aligned at the document level are provided by multilingual governments and agencies such as the Canadian Hansard and the United Nations. Mining the web for parallel documents from multilingual websites is also a potential source.

An example of a simple technique for automatic identification of parallel web sites is STRANDS (Resnik, 1999; Resnik and Smith, 2003) which first locates possibly parallel web sites; then generates candidate pairs of parallel web pages; and finally applies a structural filter to the candidate set. An alternative to build parallel corpora is to extract them from comparable corpora as done in (Fung and Cheung, 2004a; Fung and Cheung, 2004b).

1.3.2 Sentence Alignment

Sentence alignment is of ever-increasing utility with the advancement of corpus-based computational linguistics. Many applications nowadays rely on parallel sentences as input to their processing toolchain. Text is not always translated sentence by sentence. Long sentences may be broken up, or short sentences may be merged. There are even some languages where the clear indication of a sentence end is not part of the writing system (for instance, Thai). Figure 1.6 shows an example of sentence aligned bitext.

Many sentence alignment methods have been proposed in the literature. Some are based on the length of sentences (Brown, Lai, and Mercer, 1991; Gale and Church, 1993). Kay and Röscheisen (1993) propose an iterative algorithm that uses basic features such as spelling

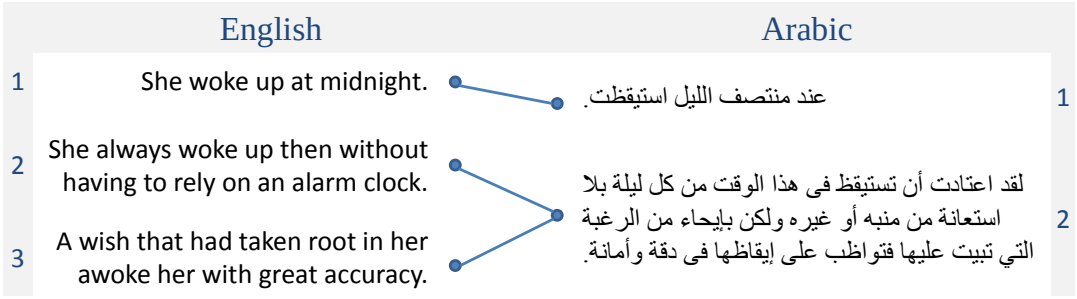


Figure 1.6: *A bitext aligned at the sentence level. (N. Mahfouz (Bayn al-Qasrayn) Palace Walk (1962)).*

similarity and word co-occurrences. Geometric (Melamed, 1996a) and pattern recognition (Melamed, 1999) approaches have also been used to identify the alignments. Chang and Chen (1997a); Melamed (1997) apply line detection methods from image processing. In addition to basic statistics, lexical information proved helpful (Chen, 1993; Dagan, Church, and Gale, 1993; Utsuro et al., 1994; Wu, 1994; Langlais, 1997) and in more recent work of (Kueng and Su, 2002; Moore, 2002). Singh and Husain (2005) present a comparison between different sentence alignment methods.

Since sentence alignment is dominated by one-to-one mappings without crossing links (monotonic), simple cues such as length correlations, and incomplete lexical constraints are often sufficient to perform reasonably well. However, in many cases, a previous alignment of larger textual units (paragraphs, sections, chapters) is useful to improve alignment quality and speed. Sentence alignment may also benefit from alignment of smaller units such as word alignment (Kay and Röscheisen, 1993).

1.3.3 Sub-sentential Alignment

The focus of this dissertation is on alignment at a sub-sentential level: words, phrases clauses and expressions. The input bitext to the alignment algorithm is a set of parallel sentences[5], generally aligned one-to-one. The output is a sub-sentential alignment[6].

The first processing step is to tokenize the sentence into a sequence of distinct tokens (or words[7]). Such tokenization must be adapted to the translation direction and to the language pair at hand. The input parallel sentence is then represented as $(\mathbf{f}, \mathbf{e})$, where the vector $\mathbf{f} = (f_1, \ldots, f_N)$ represents a source language sentence composed of N words and the vector $\mathbf{e} = (e_1, \ldots, e_M)$ similarly represents a target language sentence composed of M words.

The output alignment falls in one of two major categories: *word* and *phrase* alignment, depending on the size of the sub-sentential units involved in the alignment. Phrases may be restricted to match some linguistic definition as in chunk alignment.

1.3.3.1 Word alignment

A word alignment between two parallel sentences $(\mathbf{f}, \mathbf{e})$, of lengths N and M respectively, refers to the set of links (pairing) between single word *positions* between the two sentences. Let $\mathcal{N} = \{i : 1 \leqslant i \leqslant N\}$ be the set of source positions and $\mathcal{M} = \{j : 1 \leqslant j \leqslant M\}$ be the set of

[5]Or parallel text chunks, depending on the application in mind (Deng, Kumar, and Byrne, 2007)

[6]From now on, the term "alignment" refers to a sub-sentential alignment unless it is stated otherwise.

[7]In this context a word is stripped of its linguistic meaning and only represents a sequence of non-blank characters.

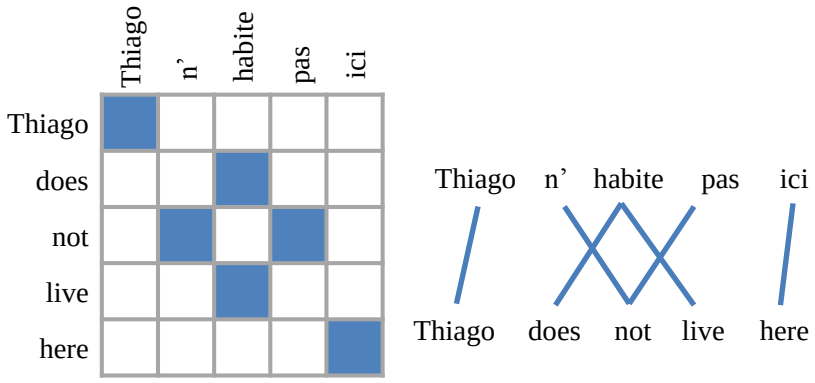


Figure 1.7: *Example of word alignment. Two equivalent representations of a word alignment are given: matrix (left) and links (right).*

target positions. The word alignment is defined as:

$$\mathbf{A} = \{(i, j) : i \in \mathcal{N} \text{ and } j \in \mathcal{M}\}. \tag{1.1}$$

A link $(i, j) \in \mathbf{A}$ represents a translation relation between the associated words at the given positions. Matching is only possible between single word positions, meaning that only single words can be *explicitly* put in a translation relation.

Word alignment is typically non monotonic with crossing links, and not bijective, including many-to-many associations. Unaligned words are authorized: not all positions should be covered by a link[8]. Figure 1.7 gives an example of word alignment.

1.3.3.2 Phrase alignment

The word alignment can be generalized and instead of allowing only single words to be linked, a phrase alignment allows for multiple words to be grouped together and linked as if they would represent a single text unit called a *segment* or a *phrase*[9]. Phrases may be contiguous or may contain gaps as in the French "`ne * pas`", hence called *gappy* phrases.

A segment can be represented by a *coverage set* containing the corresponding word indices. The set $\mathbf{p}$ characterizes a source segment and $\mathbf{r}$ a target segment. A segment pair $(\mathbf{p}, \mathbf{r})$ is an association between a source and a target segment. Unlike links in word alignment, a segment pair can explicitly represent a many-to-many translation relation. A phrase alignment is then defined as:

$$\mathbf{A} = \{(\mathbf{p}, \mathbf{r}) : \mathbf{p} \subseteq \mathcal{N} \text{ and } \mathbf{r} \subseteq \mathcal{M}\}. \tag{1.2}$$

Alternatively, and in many cases more conveniently than coverage sets, segments can be identified by their spans instead of their index sets, such that $span(\mathbf{p}) = (s, t)$ where $s = min(\mathbf{p})$ is the start position and $t = max(\mathbf{r})$ is the end position. Excluded indices (gaps) are kept in a separate set g. $span(\mathbf{p})$ bounds the sentence words $(f_s, \ldots, f_t)$ where $1 \leqslant s \leqslant t \leqslant N$. The target segment can be defined similarly.

Segments can contain gaps and overlap arbitrarily or in some nested structure. However, contiguous or disjoint segments might be required by some applications and such constraints might be necessary. In the literature, the term "phrase alignment" typically refers to a disjoint segment alignment. Figure 1.8 gives an example of phrase alignment.

[8] Alternatively, unaligned words can be linked to a special *null* token, added to both sentences at position 0

[9] A phrase in this context does not necessarily correspond to any linguistic definition.

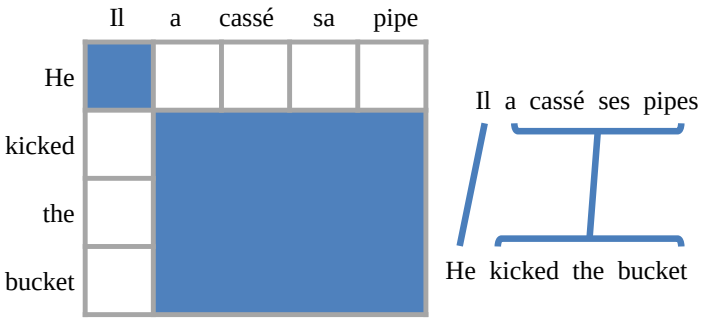


Figure 1.8: *Example of phrase alignment.*

1.3.3.3 Structure and tree alignment

Structure alignment produces a matching between grammatical constituents of a sentence pair. The segments to be aligned are obtained from constituent analysis of the sentences. Tree alignment is a special case where the output must be strictly compositional, hierarchical alignment, i. e. segments within two linked sub-trees align only to each other. The difference between word and phrase alignment and structure alignment is that the input in the former is only the sentence pair, while the input in the later is the sentence pair with its structural annotation. These alignment can be viewed as phrase alignments with additional structure constraints as we will describe in Section 1.6.1.3.

1.4 Applications

Bitext alignments at different levels of granularities have been exploited in a wide range of applications in corpus based linguistics (Véronis, 2000).

Aligned text was used to compute cross-indexing for bilingual concordances (Warwick and Russell, 1992), help language learners and bilingual readers, improve automatic translation checking tools (Macklovitch, 1994), and provide better interfaces for lexicographers, annotators and translators (Klavans and Tzoukermann, 1990). Other computer-aided translation tools and translation memories have benefited from alignment for the extraction of domain-specific translation of terminology (Gaussier and Langé, 1995; Langlais and Véronis, 1998; Langlais, Foster, and Lapalme, 2000; Kwong et al., 2002; Bourdaillet et al., 2009; Esplà, Sánchez-Martínez, and Forcada, 2011).

Alignments were used in automatic acquisition of word dictionaries from parallel corpora (Melamed, 1996b), query expansion in monolingual information retrieval (Xu, Fraser, and Weischedel, 2002; Riezler et al., 2007), cross-language information retrieval (Wang, 2005), cross-lingual syntactic learning (Yarowsky, Ngai, and Wicentowski, 2001; Smith and Smith, 2004; Hwa et al., 2005), synonym acquisition (Plas and Tiedemann, 2006), WordNet-like lexico-semantic relation extraction (Diab, 2004; Sagot and Fišer, 2008), paraphrases (Pang, Knight, and Marcu, 2003; Quirk, Brockett, and Dolan, 2004; Bannard and Callison-Burch, 2005), word sense disambiguation (Resnik and Yarowsky, 1997). Even limitations of word alignment models turned out to be helpful in identifying non-compositional idiomatic expressions (Villada Moirón and Tiedemann, 2006).

In machine translation, alignment use is not reserved to statistical approaches, it can be used in *Example-Based Machine Translation (EBMT)* (Nagao, 1984) for chunk alignment and building translation bilexica, and even in non-statistical approaches for lexicon extraction and rule induction. Alignments of various granularity can be exploited by different applications as reproduced in Figure 1.9 (adapted from (Tiedemann, 2011)).

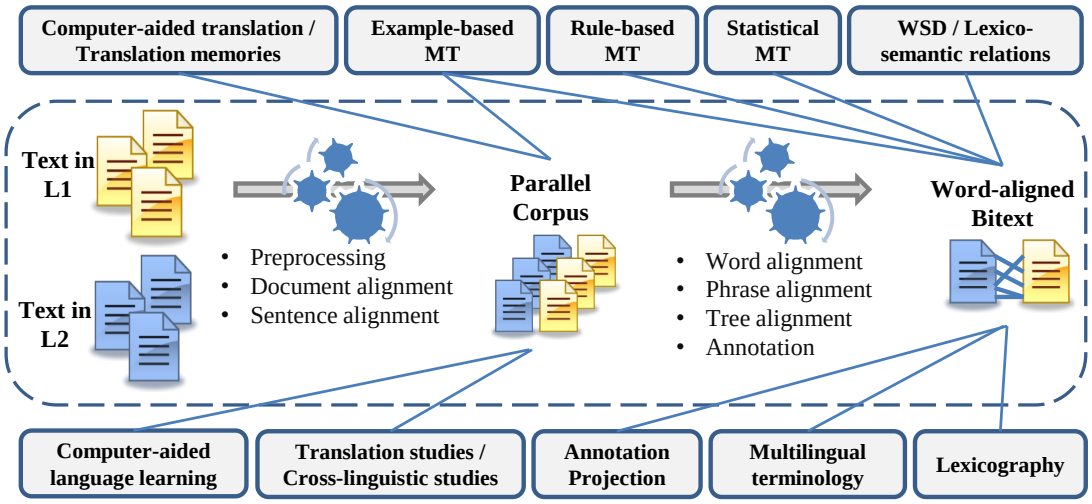


Figure 1.9: *Application areas of aligned parallel corpora.*

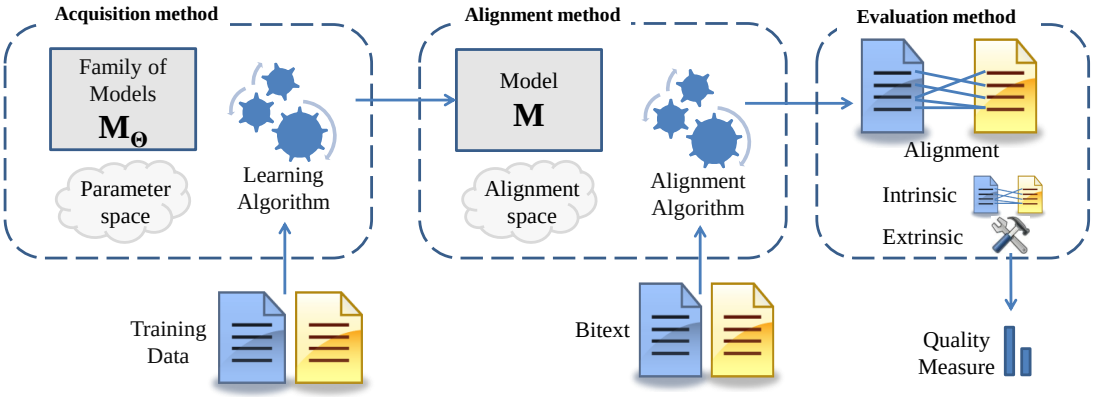


Figure 1.10: *A framework for solving the alignment problem.*

Alignments can also be used in "interlingual" translation where the source and target texts belong to the same language; this includes rewording and paraphrase, and in intersemiotic translation from verbal to non-verbal sign as music or image. Moreover, many alignment algorithms can be straightforwardly generalized to non-linguistic applications where words, phrases and sentences can be substituted by other kinds of tokens, segments and sequences. Alignment algorithms are found in bioinformatics, for pairwise and multiple DNA and RNA sequence alignment for genomes annotation (Sharma, 2008), in handwritten recognition systems to align text images and their transcripts (Fischer et al., 2011). Similarly, alignment techniques are used in ontology and XML schema matching (Euzenat and Shvaiko, 2007).

1.5 A Generic Framework for Alignment

We have described the bitext alignment problem from a linguistic angle, and discussed its main characteristics and difficulties. We now turn our attention to applied computational linguistics and present a framework in which a solution to the alignment problem can be described. This framework is depicted in Figure 1.10. A generic representation of statistical modeling in NLP is discussed in (Nivre, 2002).

The *alignment method* contains a *model* M which is simply a set of rules that represents the facts of the world relevant to the alignment problem. The model is accompanied by an algorithm which, given an instance of the problem, consults the model to find a solution efficiently. The model can be "rule-based" or "statistical" where the rules are probabilized. The algorithm is typically deterministic but can be stochastic, especially when computing exact solutions to the alignment problem is intractable.

The *acquisition method* part is the factory where the model is constructed. The application model M is instantiated in this factory from a parameterized model $M_{\boldsymbol{\theta}}$ by providing values to the parameters $\boldsymbol{\theta}$.

At last, the *evaluation method* is concerned with assessing the performance of the components of the framework. Each alignment approach is characterized by a set of decisions made at different points in the framework. Such decisions vary along five axes:

- **Input and output spaces.** Both the input and the output of the framework are structured objects. *The input* is for instance a pair of sentences $\mathbf{x} = (\mathbf{f}, \mathbf{e}) \in \Sigma^* \times \Lambda^*$ where Σ and Λ are the vocabularies of the two languages. The input may be more complex than plain sentences. Alignment may be required between nodes in the linguistic structures of the sentences and not only between their units.

  The output is an alignment representing translational correspondences between source and target units. An alignment is denoted by[10] $\mathbf{A} \in \mathcal{A}$, where $\mathcal{A}$ is the set of all possible alignments for the input $\mathbf{x}$. The alignment space is huge and usually needs to be restricted according to a set of *constraints*. A model of *translational equivalence* between a pair of sentences implements such constraints and enumerates all possible alignment structures.

- **Search.** The alignment predictor $h \in \mathcal{H}, h : \Sigma^* \times \Lambda^* \to \mathcal{A}$ maps the input space onto the output space. For a given pair of sentences $\mathbf{x}$, the predictor outputs their alignment $h(\mathbf{x}) = \hat{\mathbf{A}}$. The predictor usually uses an internal **cost function** $\omega : \Sigma^* \times \Lambda^* \times \mathcal{A} \to \mathbb{R}$, used to rank the alignment candidates in the output space. Finding the best alignment is then formulated as a **search** problem in the alignment space:

$$\hat{\mathbf{A}} = h(\mathbf{x}) = \min_{\mathbf{A} \in \mathcal{A}} \omega(\mathbf{x}, \mathbf{A}). \tag{1.3}$$

  The search space $\mathcal{A}$ is typically very large. It includes all possible subsets of the Cartesian product between the source and the target sentences. Therefore, an exhaustive brute-force enumeration of all alignment is not tractable. However, under some independence assumptions, such as the alignment of each word depends only on its neighbors, the minimization can be performed efficiently using dynamic programming (DP). In many cases the size of the search space is prohibitively large which implies the use of heuristic search. A detailed discussion of the search space follows in Section 1.6.

- **Model and parameterization.** The cost function is defined according to a model $\mathbf{M}$. The model is parameterized with a set of parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ which are often learned from the data. The model is a set of rules used to compute the cost function.

- **Training data.** The majority of statistical alignment techniques rely on a parallel corpus for parameter estimation. **Unsupervised** learning is required if the the parallel corpus is aligned only at the sentence level $\mathcal{D} = \{\tilde{\mathbf{e}}_k, \tilde{\mathbf{f}}_k\}_{k=1}^{\tilde{N}}$. **Supervised** learning can be used if the training corpus is augmented with sub-sentential alignment annotation $\mathcal{D} = \{(\tilde{\mathbf{e}}_k, \tilde{\mathbf{f}}_k, \tilde{\mathbf{A}}_k)\}_{k=1}^{\tilde{N}}$. A mid-ground scenario is the **semi-supervised** learning where some training examples in the corpus are annotated while others are not.

[10] It is worth noticing that an alignment $\mathbf{A}$ of $\mathbf{x}$ may have different definitions depending on the alignment granularity (word, phrase, ...). See Section 1.3 for examples.

- **Parameter estimation.** Learning parameters from the data is usually cast as a **minimization** problem of a **loss** function. The loss is approximated by the **empirical risk** estimated using the sample distribution of the training data:

$$\hat{\boldsymbol{\theta}} = \min_{\boldsymbol{\theta} \in \mathbb{R}^d} \frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} loss(\tilde{\mathbf{x}}_i, \tilde{\mathbf{A}}_i; h_{\boldsymbol{\theta}}). \tag{1.4}$$

  The loss function $loss(\mathbf{x}, \mathbf{A}; h_{\boldsymbol{\theta}})$ measures how bad is the alignment predictor, parameterized by $\boldsymbol{\theta}$, for an input $\mathbf{x}$ given that the correct alignment is $\mathbf{A}$. This implies that parameter estimation naturally involves a search in the alignment space since the output of the predictor has to be computed for each training instance as in Equation 1.3.

1.6 Alignment Space and Constraints

The set of alternative alignments is called the *search space* or the *hypothesis space*. An automatic alignment algorithm has to include a mechanism to explore the search space and to decide which alignment is best. If $\mathcal{B}$ is the set of *all* segment pairs that can be defined on a parallel sentence, then the search space is its *power set* denoted $\mathcal{A} = \mathcal{P}(\mathcal{B})$. Therefore, the size of the search space is $2^{|\mathcal{B}|}$, with $|\mathcal{B}|$ = the number of source segments $\times$ the number of target segments.

The size of $\mathcal{B}$ changes according to the constraints applied on the alignable segments. In the case of word alignment, segmentation is fixed and possible source (target) segments correspond to source (target) words. Hence, the size of the word alignment search space is restricted to $2^{N \times M}$. While in unconstrained phrase alignment we have 2^N source segments and 2^M target segments, meaning that the size of the search space is $2^{2^{N+M}}$.

In order to select one alternative, the algorithm should be able to quantitatively evaluate and compare alignments. This is usually done via a cost function $\omega : \mathcal{A} \rightarrow \mathbb{R}$. How such scoring function is actually calculated to reflect properly the translation relation is decided by the *alignment model* that is chosen for a particular task. Alignment scores are typically derived from distributional features, correlations and interactions between individual links.

Several problems require to explore the alignment search space. For instance, the alignment problem itself which consists of the search in the hypothesis space for the optimal, "Viterbi" alignment $\hat{\mathbf{A}}$ in the sense of the cost function, as described in Equation 1.3. Another problem is the computation of a weighted count for a specific segment pair under all alignments that permit it, called the *expectation* problem:

$$\varepsilon(\mathbf{p}, \mathbf{r}) = \sum_{\{\mathbf{A} \in \mathcal{A} \,:\, (\mathbf{p}, \mathbf{r}) \in \mathbf{A}\}} \omega(\mathbf{x}, \mathbf{A}). \tag{1.5}$$

Constraints on the hypothesis space of alignment are necessary for two main reasons. First, the unrestricted search space is prohibitively too large to be exhaustively explored efficiently. Therefore, a set of constraints is used to shrink the size of this space. Second, constraints represent prior knowledge about the correspondence structure, and bias the alignment towards some desired properties.

Constraints are expressed as limitations on segments (segmentations) and on their alignments (segment matching). A detailed discussion can be found in (Wu, 2010; Tiedemann, 2011).

1.6.1 Segment Constraints

Segment constraints restrict the set of source and target segments that can be aligned, independently from each another and from the set of matchings that can be established between them afterwords. For instance, only segments that correspond to linguistically

motivated units may be allowed. Without any restrictions, there are 2^N segments in a sentence of length N.

1.6.1.1 Contiguity constraints

Segments can be constrained to a maximum number of gaps and gap size. For example, in the English sentence "I do not want to play anymore", the segment indexed by $\mathbf{p} = \{1, 3, 7\}$ corresponding to "I * not *** anymore" has two gaps, the first is of size one and the second is of size three. A segment p is *contiguous* when it has no gap, that is when: $\forall i$ s.t. $min(\mathbf{p}) \leqslant i \leqslant max(\mathbf{p}) : i \in \mathbf{p}$. The contiguous segment constraint reduces the number of segments in a sentence of length N from 2^N to $\frac{1}{2}N(N+1)$, thereby pruning much of the search space.

1.6.1.2 Length constraints

Length constraints specify the maximum number of source and target word tokens in allowed in a segment pairs. Such constraints are applied to reduce the size of the hypothesis space. If no segment can exceed the length of n, the number of segment extracted from a sentence of length N becomes $(n+1)\binom{N}{n}$. Using the length constraint in conjunction with using only contiguous segment is widely used in applications such as machine translation. The number of contiguous segments of maximum size n in a sentence of length N shrinks down to $\frac{1}{2}(n+1)(2N-n)$.

While long segments capture wider context that short segments, they tend to be much less frequent, and can be decomposed into shorter, and typically more frequent segments. Therefore the length constraint forces the alignment algorithm to focus on short segments.

1.6.1.3 Structural constraints

Structure constraints provide ways to control the overlap between segments. Many alignment algorithms consider one fixed *disjoint* segmentation of each monolingual sentence. A disjoint segmentation of a sentence contains segments that cover the entire sentence and do not overlap.

Authorized segments can be enriched with compositional (hierarchical) ones that result from joining neighbor disjoint phrases to form a tree. Monolingual syntactic parsers can be used to produce a grammatical tree where segments correspond to the grammatical phrases.

Word alignment implies disjoint fixed segmentation (at word boundaries) while alignment with structural constraints on both sides are referred to as *tree alignment*.

1.6.2 Alignment Constraints

Alignment constraints are applied on the set of links between authorized segments.

1.6.2.1 Structural constraints

In tree alignment, the tree structures of the the input sentences are pre-computed using parsers, then the alignment algorithm matches the nodes of these trees. Alternatively, the parsing steps may be omitted, and it is up to the alignment algorithm to explore the possible structures and output the structures and the alignments. Therefore, instead of using segment constraints to pre-selecting the structure, a structural constraint on the output alignment is applied. Similarly, the alignment algorithm may be constrained to output a disjoint segmentation along with the alignment instead of fixing the segmentation *a priori*.

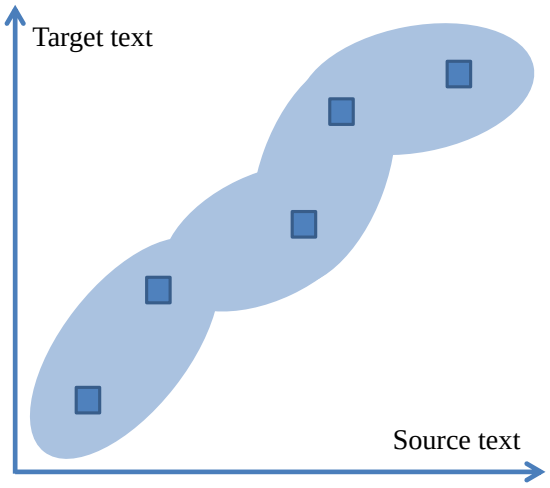


Figure 1.11: *Guiding alignment.*

1.6.2.2 Range constraint

Range constraints include several types of constraints of which the main idea is to restrict the permissible alignment links to some confined region in the hypothesis space.

One such example is the *monotonicity constraints* which require paired source and target segments to occur in the same order in both aligned sentences. Monotonicity constraints are rooted in automatic speech recognition applications where acoustic waveforms need to be aligned monotonically to transcriptions. Similarly, Optical Character Recognition (OCR) to text alignment and genome sequence alignment are monotonic.

Since crossings between links are not allowed under these constraints, choosing to match two particular segments, as an *anchor constraint*, divides the parallel sentence into two disjoint ones that could be aligned separately in a recursive way. Monotonicity and anchoring constraints are more helpful for document and sentence alignment than for sub-sentence alignment where the assumptions behind them become unreasonable.

Guiding constraints are applicable when a rough alignment already exists. A window of given size around "guide links" specify the region of allowed links in the alignment matrix, in which more accurate alignment can be looked for. This is shown in Figure 1.11. Guiding constraints are suitable for iterative algorithms which start with a first estimate of the alignment and use it to seek enhancement in subsequent iterations.

Distortion constraints are also frequently used to limit the maximum distance between the positions of the aligned segments, calculated as the distance from the diagonal of the alignment matrix.

1.6.2.3 Functional constraints

Alignment relation can be represented as a function mapping elements between the two sets of source and target segments, by designating one set as the domain and the other as the co-domain. In light of such representation, some alignment constraints can be expressed as function properties. Three such constraints are represented in Figure 1.12.

Function representation imposes that every source segment should be aligned to exactly one target segment, resulting in *many-to-one* constraints, or similarly *one-to-many* by exchanging the domain and the co-domain. Naturally, these constraints lead to *asymmetric* models in which the alignment direction is important. *Injective*, or *one-to-one* constraints require that

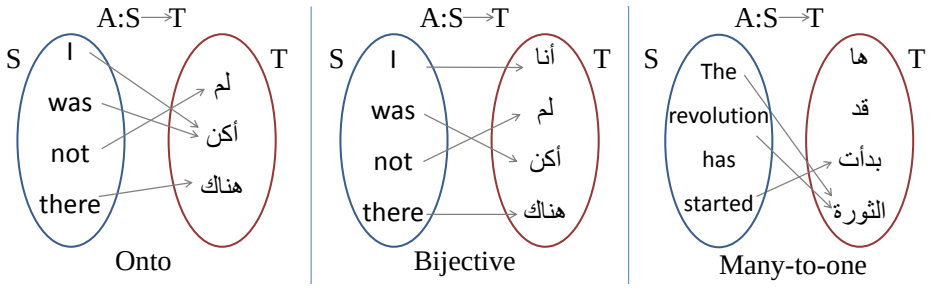


Figure 1.12: *Some of the alignment functional constrains.*

each target segment is mapped to at most one source segment. *Surjective* constraints guaranty that all target segments are aligned.

1.6.2.4 Bijectivity constraints

Bijectivity means that the alignment is one-to-one between disjoint segments that cover the sentences. Many alignment algorithms, as well as the phrase-based[11] decoding framework (Och and Ney, 2004) operate under bijective constraints.

Bijective constraints yield a bisegmentation (García-Varea et al., 2005; Wu, 2010). The space of bijective alignment is sometimes called the *permutation space* (Cherry and Lin, 2006a) as the number of possible alignments is reduced to $s!$, where s corresponds to the number of segments.

In many cases, aligning all segments is an unrealistic assumption and we may wish that some segments remain unmatched. Partial mapping can also be achieved by adding artificial empty units (or `null` tokens), to algorithms that require a full mapping.

1.7 Evaluation Methods

At the end, an *evaluation method* is needed to assess the performance of the alignment system. After applying the acquisition algorithm (M_Θ, A_1) to some corpus C to construct the model M, the application algorithm (M, A_2) is used to align some bitext. These alignments are then evaluated either intrinsically by comparison to a manual alignment or extrinsically by evaluating their performance when plugged into an external application.

Quantitative evaluation of alignment quality is a difficult task, basically for the same reasons that make the task of alignment itself difficult.

1.7.1 Intrinsic Measures

These methods estimate how much an alignment succeeds in accomplishing the task set by its definition, namely finding *all* bilingual correspondences between source and target phrases.

1.7.1.1 Alignment Error Rate (AER)

For this purpose, *gold* alignments are established by human annotators on a test set, and used as a reference for comparison. In order to simplify the annotation task, only word-level links are typically used.

Alignment Error Rate (AER) (Och and Ney, 2003), thus, measures the quality of automatic word alignments against these gold alignments. To confront the alignment difficulties,

[11] We use the wording "segment" to refer to the same entity referred to by "phrase" in phrase-based translation.

annotators are instructed to follow strict guidelines that provide conventional solutions (Melamed, 1998). For example, annotators may be asked to never align functional words that do not have counterparts instead of taking arbitrary decisions about them.

Although non-compositional phrases cannot be aligned on the word-level, it is the only level on which the AER can be calculated. To resolve this mismatch, annotators are allowed to produce two sets of links. The set $\boldsymbol{S}$ that contains *sure* points aligned with no ambiguity; and the set $\boldsymbol{P}$ that contains in addition to sure points, *probable* points used where no one-to-one correspondence is possible. For a given alignment $\mathbf{A}$, *precision*, *recall* and AER are defined as:

$$\text{Precision}(\mathbf{A}, \boldsymbol{P}) = \frac{|\mathbf{A} \cap \boldsymbol{P}|}{|\mathbf{A}|} \qquad \text{if } |\mathbf{A}| > 0, \text{ and 1 otherwise;} \tag{1.6}$$

$$\text{Recall}(\mathbf{A}, \boldsymbol{S}) = \frac{|\mathbf{A} \cap \boldsymbol{S}|}{|\boldsymbol{S}|} \qquad \text{if } |\boldsymbol{S}| > 0, \text{ and 1 otherwise;} \tag{1.7}$$

$$\text{AER}(\mathbf{A}, \boldsymbol{P}, \boldsymbol{S}) = 1 - \frac{|\mathbf{A} \cap \boldsymbol{P}| + |\mathbf{A} \cap \boldsymbol{S}|}{|\mathbf{A}| + |\boldsymbol{S}|} \qquad \text{if } |\mathbf{A}| + |\boldsymbol{S}| > 0, \text{ and 0 otherwise.} \tag{1.8}$$

Precision measures the overlap between the set of hypothesized links and the set of links annotated as "possible" in the gold alignment. Precision is at its maximum when no links are hypothesized ($\mathbf{A} = \emptyset$), and decreases only when links that are not neither "possible" nor "sure" are added.

Recall measures the percentage of "sure" links that are found in the hypothesized alignment. Recall is at its maximum when the alignment contains all possible links, and decreases only when "sure" links are removed.

When no distinction between "sure" and "possible" links is made ($\boldsymbol{S} = \boldsymbol{P}$), 1-AER reduces to to the standard *F-measure*.

It is argued in (Fraser and Marcu, 2007b) that when such distinction is present ($\boldsymbol{S} \subset \boldsymbol{P}$), AER does not penalize unbalanced precision and recall contrarily to the F-measure. Therefore it is possible to maximize AER by favoring precision over recall, which can be done by simply guessing very few alignment links. This mathematical formulation of AER leads to strong biases which questions its use as the reference metric for alignment quality. The same was previously observed in (Goutte, Yamada, and Gaussier, 2004).

1.7.1.2 Balanced F-measure

An F-measure without the "sure" and "possible" distinction is presented in (Fraser and Marcu, 2007b):

$$\text{F-measure}(\mathbf{A}, \boldsymbol{S}, \alpha) = \frac{1}{\frac{\alpha}{\text{Precision}(\mathbf{A}, \boldsymbol{S})} + \frac{(1-\alpha)}{\text{Recall}(\mathbf{A}, \boldsymbol{S})}}, \tag{1.9}$$

where $0 \leqslant \alpha \leqslant 1$ controls the trade-off between precision and recall, which can be fine-tuned for a specific task by varying α. They show that this measure has a better ability to capture intrinsic alignment quality, and is also reflected by better extrinsic prediction of alignment performance in external applications such as translation.

1.7.1.3 Other word-level measures

Several efforts have been made along this research axis where alignment are compared against word-level gold standards using some distance metric. The focus of these approaches was to find weighting schemes of word links that reflect the many-to-many word correspondence in non-compositional translations. The distinction between "sure" and "possible" links in AER is introduced by (Och and Ney, 2003) to help properly evaluate non-compositional links,

which is criticized in (Fraser and Marcu, 2007b). In (Melamed, 2000; Davis, 2002) the sum of weights of all links to a word should be a constant to avoid overweighting such links. A simple link precision/recall metric is developed in (Ahrenberg et al., 2000) to evaluate the alignment of multiple English words to the large compound words in Germanic languages.

These various methods of comparisons are based solely on one aspect of the alignments, namely the present links. Additional characteristics of the alignment are investigated in (Guzman, Gao, and Vogel, 2009), and compared against those of hand-aligned gold standards. The idea is to use a richer representation of the compared alignments, so as to get a deeper understanding of their differences and similarities. These characteristics have the form of summary information concerning either present links, such as the total number of links in an alignment and its average link density; or missing links such as the number of unaligned words and nonalignment rate. While these statistics characterize different aspects of an alignment, there exists no measure that uses them quantitatively.

All these evaluation metrics share the need for gold alignments. This can be avoided by working out a *confidence measure* (Huang, 2009) by simple combination of posterior probabilities of individual links, under some alignment model. This confidence measure is showed to be correlated with the standard F-measure, which makes it useful when no gold alignments are available.

The main issue with word-level evaluation metrics is the difficulty to deal with non-compositional phrase alignment. A single non-compositional correspondences are usually annotated with many-to-many alignments. However, word-level metrics treat these links individually as in one-to-one correspondence. This mismatch is addressed in phrase-level measures.

1.7.1.4 Phrase-level measures

In order to solve the non-compositional phrase evaluation problem some measures consider gold standards that include linked units at the phrase-level.

Some approaches to measure the alignment quality do not involve using a gold standard word alignment, but instead build a translation lexicon from the whole alignment. Wu and Xia (1995) sample the translation lexicon built from the alignment and uses both manual and automatic filters to measure precision. Melamed (2000) measures probability weighted precision manually, that is then used to estimate probability weighted recall. Alignments can be compared to entries in a dictionary as in (Koehn and Knight, 2002), or to reference bilingual lexicon (Lardilleux, Gosme, and Lepage, 2010). The disadvantage of these methods is that phrase-level gold standards are not easily obtainable; and that phrase pairs are evaluated out of the context from which they were extracted.

A measure proposed in (Ayan and Dorr, 2006b) called Phrase Consistency Error Rate (PCER) attempts to remedy both of these problems, and avoids overweighting links in non-compositional units. Similar to F-measure, PCER incorporates sentence-level context and equally weights precision and recall over phrases extracted from the hypothesized alignment with respect to phrases extracted from the gold alignment.

As in the word-level case, phrase pairs can be compared to reference ones according to additional characteristics (Guzman, Gao, and Vogel, 2009). Interesting statistics would characterize the *singleton* phrase pairs, length of the involved phrases, and unaligned words inside them, called *gaps*.

1.7.2 Extrinsic Measures

The final judgment of the quality of a given alignment is made in the context of its final application. Instead of comparing alignment to hand-aligned data, extrinsic metrics measure the impact of the alignment on the output quality of the application. This is done by holding all the components of the application unchanged, and varying only the alignments.

In statistical machine translation as an application of word alignment, any translation quality metric can be used as an extrinsic alignment quality measure. Most widely used in practice are n-gram matching metrics such as BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), or TER (Snover et al., 2006a).

Finding an automatic approach for evaluating the translation quality that is correlated with human judgment is a highly active research field. A comprehensive discussion of different categories of such approaches can be found in (Koehn, 2010).

1.7.3 Correlation

In order for any alignment quality measure to be useful for some external application, it ought to be a good predictor of the performance of the final application. Therefore, high correlation between an intrinsic and an extrinsic measure guarantees that any improvement in quality measured by the former carries to the later. This is important for many applications, such as machine translation, since calculating the intrinsic measure is much less expensive than calculating the translation quality measure. Such correlation allows the alignment algorithm to use scoring functions that predict translation quality without involving the irrelevant components of the external translation system.

Unfortunately, the existence of such correlated measures in machine translation is a highly debatable subject, and completely contradictory conclusions can be drawn in varying circumstances. This is not surprising since intrinsic measures that compare alignments to gold standards lack the flexibility to consider different alignment properties for different translation tasks (e.g. different language pairs and different training corpora sizes), and different downstream translation approaches (Fraser, 2007; Lopez, 2008b).

1.8 Summary

Bitext alignment is the problem of finding correspondences between a text in the source language and its translation in the target language. The goal is to explain the coarse translation relation in the bitext, in terms of finer units at different levels of granularity, such as documents, sentences and words. Translation, and therefore the alignment, is rarely monotonic or word-for-word. This is mainly due to two reasons. First, languages differ in many ways, including morphology, syntax, semantics and pragmatics. Therefore, concepts may be conveyed using variable number of words and with different order across languages. Second, the translation strategy varies from literal, which preserves the form of the original text and its meaning; to free which is concerned only with meaning. For distant languages, and free translation, alignments become coarse-grained with large differences in relative word order. Typically, alignment is performed separately for each level of granularity, starting from the document level down to the sub-sentential level. The focus of this dissertation is the word and phrase alignment.

Nowadays, the body of translated texts is increasing steadily and many applications of bitext alignment are emerging. For such applications, the presence of reliable automatic alignment methods is vital. For this purpose, we described a data-driven approach based on statistical modeling, and discussed several particularities of the alignment task. The number of possible alignments of a bitext is typically very large. Therefore, constrains are applied on the ways of segmenting each text, for instance, the number of words per segment may be constrained, or segments may have to conform to a hierarchical structure; and the ways of linking these segments, for instance only monotonic alignments may be allowed or crossing within a certain range, the number of links per segment may also be constrained, etc. We have also discussed an important aspect of the alignment framework which is the existence of automatic quality metrics. Widely used intrinsic metrics compare the output of the alignment framework to manual alignments, and combine recall and precision criteria as in AER and

F-measures. However, most of such metrics function on the word level and may not be capable of capturing equivalence for larger segments. Moreover, the alignment quality is ultimately evaluated in the context of an external application such as machine translation using extrinsic metrics such as BLEU, which is not necessarily correlated with the intrinsic quality. This is a problem, because extrinsic metrics are typically more computationally expensive than intrinsic metrics.

CHAPTER 2

Alignment Models

In Chapter 1 we have described several aspects of the bitext alignment task and have presented a generic framework for solving the alignment problem. We also have briefly described document and sentence alignments as they are useful for many applications. They also constitute a starting point for the task of sub-sentential alignment which is the focus of this chapter. We divide the sub-sentential alignment models into word-based based and phrase-based. Word-based models use a strict constraint on the length of the alignable units and only consider words. This constraint reduces the alignment search space. However, it does not match well the nature of translation for many language pairs and translation strategies. Therefore, phrase-based models allow groups of words, called phrases, to align as a whole. For each of word and phrase models, we explore the literature for concrete instantiations of the alignment framework and discuss advantages and weaknesses of each approach and how they relate to each other.

The first word-based approaches (Brown et al., 1993) modeled the alignment as hidden variable in the translation process, using a generative joint model. The model explains how the words on one side of a parallel sentence are generated from the words on the other side. Since each target word is generated from one source word, this modeling results in asymmetric, one-to-many alignments. Under such a generative model, the word alignments are obtained as a by-product of training the translation models. However, modeling the alignment variable directly is more advantageous, if the model is to be used only for alignment prediction. This is the case for discriminative approaches that emerged later, which kept the same asymmetric formulation but modeled only the alignment. Discriminative models facilitate the incorporation of additional features and allow to benefit from available manual alignments. Separating the alignment from the generative translation model opened the door to many alternative parameterization of the model, including non-probabilistic linear models and heuristic approaches.

The major limitation shared by all these models is their asymmetry and restriction to one-to-many alignments. The shift from word- to phrase-based translation models as well as new emerging applications requires symmetric alignments. This motivated the work on symmetrization methods of two directional models, including heuristics and model-based approaches. However, a more direct and principled way is to reformulate the problem and to consider a symmetric representation of the alignment. Such symmetric models started to appear in the late nineties. Wu (1997) proposed to represents the translation equivalence in

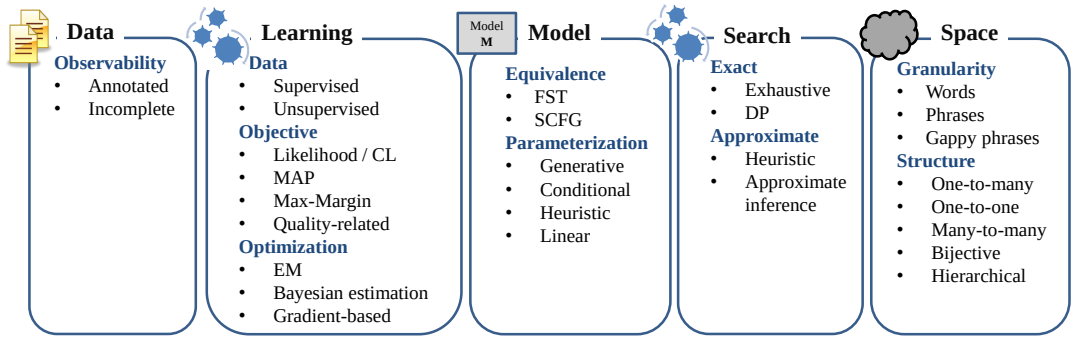


Figure 2.1: *Possible instantiation of the alignment framework.*

a parallel sentence using Inversion Transduction Grammar (ITG) which is a special case of Synchronous Context-Free Grammar (SCFG)[1]. The ITG is used to jointly parse the parallel sentence and the terminal production rules determine the aligned segments. The original formulation used a generative model to score a parallel sentence and its parse tree.Beside not being able to capture all plausible alignment patterns, the main drawback of ITG models is their computational complexity, in of the order of $O(n^6)$ where n is the length of the longest sentence, which makes pruning techniques almost inevitable. The original formulation of the ITG model allows for aligned segments to span several words, resulting in many-to-many alignments. However, this comes at the price of a huge increase in the number of parameters (terminal productions) which causes difficulties for learning the model. While this model can be used for translation (transduction), later alternatives use a discriminative model to directly score the alignment. Under such models, the ITG formalism can be seen as merely a way to constrain the space of possible alignments. More flexibility can be gained by generalizing this framework and alternative scoring functions and different constraints and pruning techniques. This is what is done in matrix modeling approaches. The alignment variable is a matrix in which each element represents the association between a source word and a target word. Obtaining the alignment consists of making a binary decision for each matrix element whether to align the corresponding words or not. This is a structured prediction problem in which all the alignment decisions are potentially influenced by one another. Since there are exponentially many configurations of the matrix, either strong independence assumptions or aggressive pruning of the space of possible alignments is required.

This chapter is organized as follows. After recalling the definition of word-based alignments in Section 2.1, we survey in Section 2.2 models that cast the alignment problem as sequence labeling problem, in which words in one sentence are labeled with the positions of their counterpart in the other sentence. Under this formulation, the simple heuristics of Section 2.2.1 have been used in the early days of alignment, before they were taken over by approaches based on machine learning techniques. This includes unsupervised generative models described in Section 2.2.2 and supervised discriminative models described in Section 2.2.3.

Symmetric approaches are discussed in Section 2.3. Sections 2.3.1 presents methods that combine two or more asymmetric alignments to obtain a symmetric one. Hence, they benefit from the advantages of sequence labeling approaches. Another class of methods which we discuss in Section 2.3.2, is based on building an alignment matrix populated with individual link costs, and then applying some algorithm on the matrix to obtain the alignment. The main issue with these methods is the difficulty to model link interactions inside the alignment. As a remedy, global discriminative models, which we discuss in Section 2.3.4,

[1] Also called "Syntax-Directed Translation Schemata (SDTS)"

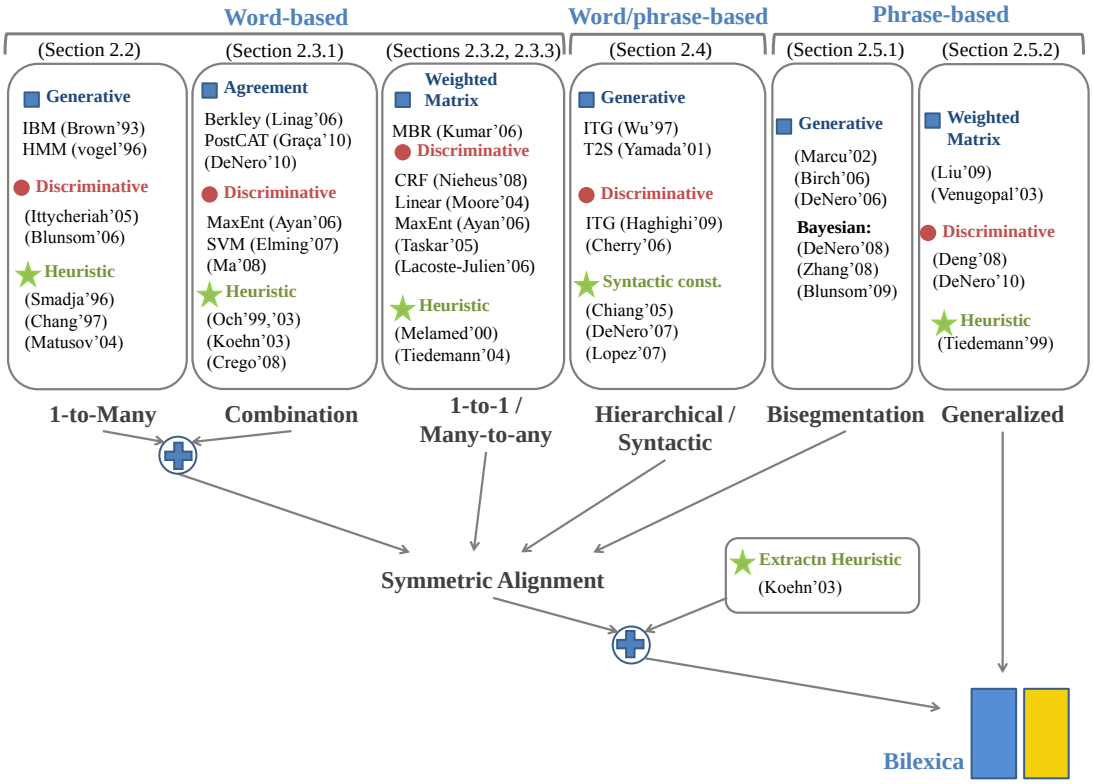


Figure 2.2: *Examples of different alignment models presented in this chapter, and how they can be used to extract translation rules for phrase-based systems called bilexica. The number of the corresponding section for each approach is shown.*

score entire alignment structures and use a search guided by this score to make predictions. Approximations are often needed to cope with the computational complexity stemming from the modeling of link interactions.

At last, in Section 2.4 we discuss syntactic and hierarchical alignment models which rely on the SCFG formalism which seems to be a good fit to model linguistic phenomena. The main advantage of these models is their ability to account for long-distance reordering without blowing up the alignment search space. Phrase-based models are discuss in Section 2.5. We distinguish between bisegmentation models (Section 2.5.1), which produce an alignment between non-overlapping phrases that covers the parallel sentence; and general phrase alignment models (Section 2.5.2) which dispense with such constraints. Section 2.6 comprises a general view of how good indicators and cues of alignment are encoded into meaningful features.

Figure 2.2 show instances of various alignment approaches presented in this chapter, and show how they can be used to extract translation rules for phrase-based systems.

2.1 Word-Based Alignment Models

Let us first recall the definition of a word alignment from Chapter 1. A word alignment between two parallel sentences $(\mathbf{e}, \mathbf{f})$, of respective length M and N, is the set of links between single word *positions* in the two sentences. Let $\mathcal{N} = \{i : 1 \leqslant i \leqslant N\}$ be the set of source positions and $\mathcal{M} = \{j : 1 \leqslant j \leqslant M\}$ be the set of target positions. A word alignment $\mathbf{A} \in \mathcal{A}$ is

defined as:

$$\mathbf{A} = \{(i, j) : i \in \mathcal{N} \text{ and } j \in \mathcal{M}\}. \tag{2.1}$$

A link $(i, j) \in \mathbf{A}$ represents a translation relation between the associated words at the given positions. Coupling is only possible between single word positions, meaning that only single words can be *explicitly* put in a translation relation. Word alignments use fixed segmentation constraints on the output space. Additional constraints are applied further by different approaches in the literature.

A word alignment is usually represented by a function $A : \mathcal{N} \times \mathcal{M} \to \{0, 1\}$ mapping the cells (i, j) in the *alignment matrix* to a binary value $A_{i,j}$ indicating whether the corresponding words are aligned or not. We should note that the number of distinct word alignments in $\mathcal{A}$ is $2^{N \times M}$, which is way too large to allow exhaustive enumeration for long sentences.

2.2 Asymmetric One-to-Many Methods

A first family of word alignment models recasts the problem as a sequence labeling task. Each target[2] word e_j is labeled with a source position $i \in \mathcal{N}$. We denote such alignment as $\mathbf{a}$ to differentiate it from the unconstrained word alignment $\mathbf{A}$. Formally, $\mathbf{a}$ is a sequence of length M of source positions. Similarly to the general case, this alignment can be seen as a function, but this time mapping positions in one sentence to positions in the other $a : \mathcal{M} \to \mathcal{N}$. The number of different possible label sequences is M^N where M is the length of the target sentence and N is the size of the label set (the source sentence positions). While this number is smaller than the general case for unconstrained alignments ($2^{N \times M}$), it is still too large to allow for exhaustive enumeration of all possible alignments.

General sequence labeling (functional, non-injective and non-surjective constraints) means that each target word is aligned to exactly one source word position, resulting in **one-to-many** alignments. Additional injectivity or bijectivity constraints result in **one-to-one** alignments. In order to allow target words not to be linked to any particular source word, the codomain of the function is usually augmented with a special *null* token at position (0). Linking a target word to this particular source position implies unalignment. A source position can be linked to zero or more target positions. No restriction on the distortion of the alignment is imposed, so arbitrary crossing links are permitted.

Sequence labeling constraints thus result in directional, *asymmetric*, many-to-one alignments. Obtaining many-to-many alignments then requires exchanging the roles of the sentences and recombining two directional alignments. Figure 2.3 show examples of two alignments in opposite directions.

2.2.1 Heuristic Alignments

The simplest method makes the alignment decisions only depend on the similarity between the words of the languages (Smadja, McKeown, and Hatzivassiloglou, 1996; Ker and Chang, 1997; Melamed, 2000). The Dice coefficient (Dice, 1945), log-likelihood ratio (Dunning, 1993) and p-value resulting from statistical significance tests are used to populate the alignment matrix with association scores $c_{i,j}$. From this association score matrix, the word alignment is obtained by applying a sequence labeling heuristic. For instance, each target word e_i is aligned to the source word with the highest association score: $a_j = \arg\max_i c_{i,j}$.

The advantage of these heuristic approaches is their simplicity. However, the choice of the scoring function is arbitrary. Furthermore, the strength of the association is overestimated unless careful adjustment are taken as pointed out by Moore (2004b). Another problem is that alignment decisions are made completely independently from one another, which is clearly

[2]Source and target are interchangeable.

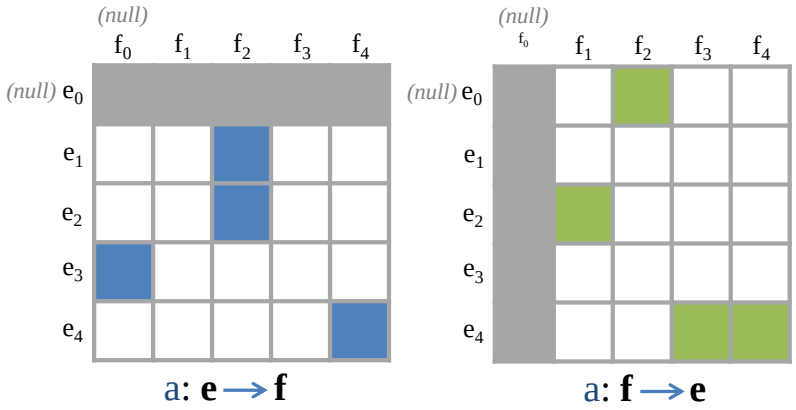


Figure 2.3: *Two directional, one-to-many word alignments for a sentence pair.*

unsuitable for many cases found in real alignments. This happens for instance when two words co-occur frequently without being translations of one another, which is called indirect associations (Melamed, 2000). Examples includes frequent words "`le / and`"; poly lexematic "`prendre la fuite / escape`"; and named entities "`Los Angeles / Los Angeles`", in this example, co-occurrence information is not sufficient to decide whether `Los` should be aligned with `Los` or with `Angeles`. Many arguments favor the use of more principled statistical alignment methods.

2.2.2 Unsupervised Generative Sequence Models

Originating from statistical machine translation (Brown et al., 1993), unsupervised translation models define the conditional lexical probability distribution $p(\mathbf{e}|\mathbf{f})$ in terms of a hidden structure representing the alignment between words in $\mathbf{e}$ and $\mathbf{f}$. The probability is re-written with the hidden alignment variable $\mathbf{a} = (a_1, \ldots, a_M)$ as follows:

$$p(\mathbf{e}|\mathbf{f}) = \sum_{\mathbf{a}} p(\mathbf{e}, \mathbf{a}|\mathbf{f}). \tag{2.2}$$

Adding the hidden alignment variable simplifies the structure of the model of $\mathbf{e}$ given $\mathbf{f}$. However, learning a model that incorporates a hidden variable is far from trivial.

We are going to consider two alternative representations, namely a Conditional Bayesian Network (CBN) and a Conditional Random Field (CRF).

2.2.2.1 Conditional Bayesian networks

The joint distribution of $\mathbf{e}$ and $\mathbf{a}$ is often decomposed using a Bayesian network, which is represented as a directed graph. Each vertex in the graph represents a random variable and each arc encodes a dependency. The network models the joint distribution of the variables in $\mathbf{e}$ and $\mathbf{a}$ conditionally on $\mathbf{f}$ which is not modeled, and therefore is referred to as a *conditional Bayesian network* (Koller and Friedman, 2009).

The joint distribution is used in decoding to find the best alignment $\mathbf{a}^*$, sometimes called the *Viterbi* alignment, given a sentence pair:

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} p(\mathbf{a}|\mathbf{e}, \mathbf{f}) = \arg\max_{\mathbf{a}} \frac{p(\mathbf{e}, \mathbf{a}|\mathbf{f})}{p(\mathbf{e}|\mathbf{f})}, \tag{2.3}$$

where $p(\mathbf{e}|\mathbf{f})$ is not used for decoding since it is the same for all values of $\mathbf{a}$. The joint distribution is parameterized with a d-dimensional vector of numerical parameters $\boldsymbol{\theta} \in \mathbb{R}^d$.

The chain rule can be applied to the joint distribution $p(\mathbf{e}, \mathbf{a}|\mathbf{f})$ which can then be rewritten in terms of individual words Conditional Probability Distribution (CPD):

$$p(\mathbf{e}, \mathbf{a}|\mathbf{f}) = p(M|\mathbf{f}) \prod_{j=1}^{M} p(a_j|e_1^{j-1}, a_1^{j-1}, f_1^N) \; p(e_j|e_1^{j-1}, a_1^j, f_1^N). \tag{2.4}$$

e_1^{j-1} is the sequence of target words from position 1 to the position $j-1$. The dependencies in each CPD can be greatly simplified by making independence assumptions.

Using this decomposition we obtain three probabilities: a **length probability** $p(M|\mathbf{f})$ which predicts the number of words in the target given the source; an **alignment probability** $p(a_j|e_1^{j-1}, a_1^{j-1}, f_1^N)$ for each position in $\mathbf{e}$, which predicts the aligned source position for a given target position given a history of all generated target words and alignments, in addition to the source; and a **lexicon probability** $p(e_i|e_1^{j-1}, a_1^j, f_1^N)$ which predicts the target word given its alignment, a history of all generated target words and alignments, and the source sentence.

Parameter estimation Estimation procedures vary depending on the actual parameterization, data observability, the objective function and the optimization methods. A multinomial parameterization of the CPDs is usually used in the alignment literature. Nevertheless, an alternative **log-linear based** parameterization is sometimes considered. The log-likelihood objective is widely used in practice and we will now present briefly the Maximum Likelihood Estimation (MLE) method in the supervised and the unsupervised cases.

- **Supervised learning.** If the training data contains the alignment annotations in addition to the parallel sentences $\{(\tilde{\mathbf{e}}_k, \tilde{\mathbf{f}}_k, \tilde{\mathbf{a}}_k)\}_{k=1}^{\tilde{N}}$, the optimization problem is the following:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{k=1}^{\tilde{N}} \log p(\tilde{\mathbf{e}}_k, \tilde{\mathbf{a}}_k|\tilde{\mathbf{f}}_k), \tag{2.5}$$

  which has, for the multinomial parameterization, a closed-form solution computed using relative frequencies of joint and marginal assignment of the random variables involved within each CPD. Other type of parameterizations, log-linear for instance, do not admit a closed form solution. When annotated data is available, discriminative supervised models are more popular since they do not model variables that are not used for alignment prediction.

- **Unsupervised learning.** A more frequent scenario is to have a large corpus of parallel sentences without alignment information $\{\tilde{\mathbf{e}}_k, \tilde{\mathbf{f}}_k\}_{k=1}^{\tilde{N}}$. In this case the hidden alignment variable is marginalized and we optimize the log-likelihood of the observable sentences. Assuming that all training sentence pairs $\tilde{\mathbf{x}}_k = (\tilde{\mathbf{f}}_k, \tilde{\mathbf{e}}_k)$ are independent and identically distributed and they sufficiently represent the entire population of translated sentences, we can write:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{k=1}^{\tilde{N}} \log \sum_{\mathbf{a} \in \mathcal{A}_{\tilde{\mathbf{x}}_k}} p(\tilde{\mathbf{e}}_k, \mathbf{a}|\tilde{\mathbf{f}}_k). \tag{2.6}$$

  which is not convex in many cases.

  The likelihood now does not decompose and the problem requires optimizing a highly nonlinear and multimodal function over a high-dimensional space which consists of parameter assignments to all CPDs. To perform this optimization, one could use a

generic optimization method such as gradient descent; or a more specialized iterative algorithm called *Expectation-Maximization (EM)* (Dempster, Laird, and Rubin, 1977), which is tailored to optimize likelihood functions. The challenge in the unsupervised case is the non-convexity of the objective function with respect to the parameters $\boldsymbol{\theta}$.

Expectation-Maximization (EM) EM iterates between calculating the posterior distributions over the hidden variables for the entire corpus $\{\mathbf{a}_k\}_{k=1}^{\tilde{N}}$ and updating the parameters $\boldsymbol{\theta}$. Starting from initial parameter settings $\boldsymbol{\theta}^{(0)}$, the algorithm repeatedly executes the following computations for $t = 0, 1, \ldots$:

- **Expectation (E-step):** Given the parameters $\boldsymbol{\theta}^{(t)}$ compute the posterior distribution over the alignment space for each training example $\tilde{\mathbf{x}}_i$ which requires to perform inference step[3]:

$$\forall \mathbf{a} \in \mathcal{A}_{\tilde{\mathbf{x}}_k},\ q_k^{(t)}(\mathbf{a}) = p_{\boldsymbol{\theta}^{(t)}}(\mathbf{a}|\tilde{\mathbf{x}}_k) = \frac{p_{\boldsymbol{\theta}^{(t)}}(\mathbf{a}, \tilde{\mathbf{e}}_k|\tilde{\mathbf{f}}_k)}{\sum_{\acute{\mathbf{a}} \in \mathcal{A}_{\tilde{\mathbf{x}}_k}} p_{\boldsymbol{\theta}^{(t)}}(\acute{\mathbf{a}}, \tilde{\mathbf{e}}_k|\tilde{\mathbf{f}}_k)}. \tag{2.7}$$

- **Maximization (M-step):** Knowing the posterior distribution of the training sentences and the alignments derive a new set of parameters $\boldsymbol{\theta}^{(t+1)}$.

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \sum_{k=1}^{\tilde{N}} \sum_{\mathbf{a} \in \mathcal{A}_{\tilde{\mathbf{x}}_k}} q_k^t(\mathbf{a}) \log p_{\boldsymbol{\theta}}(\mathbf{a}, \tilde{\mathbf{e}}_k|\tilde{\mathbf{f}}_k). \tag{2.8}$$

The expectation step is more difficult than the maximization step, since it includes inference over the set of all possible alignments. EM hill-climbs the likelihood function and is guaranteed, under some conditions, to converge to a local maximum. The quality of the obtained local maximum greatly depends on the initialization.

IBM model 1 The model in Equation (2.4) has many dependencies and cannot be reliably estimated from data. Independence assumptions are required to simplify its structures. Och and Ney (2003) presents a systematic comparison of different independence assumptions as presented in the very influential IBM models introduced by Brown et al. (1993) and the hidden Markov model introduced by Vogel, Ney, and Tillmann (1996). In the following, we discuss briefly these models which are now well known. Starting from Equation (2.4), Brown et al. (1993) consider models of increasing complexity. The first model (IBM1) makes the strongest independence assumptions and it is entirely based on lexical translations:

$$p(\mathbf{e}, \mathbf{a}|\mathbf{f}) = \frac{p(M|N)}{(N+1)^M} \prod_{j=1}^{M} p(e_j|f_{a_j}). \tag{2.9}$$

The length model $p(M|\mathbf{f})$ is simplified as $p(M|N)$: the length of the target depends only on the length of the source sentence. The alignment model $p(a_j|e_1^{j-1}, a_1^{j-1}, f_1^N)$ is uniform $\frac{1}{(N+1)^M}$, and the translation model $p(e_j|e_1^{j-1}, a_1^j, f_1^N)$ is simplified as $p(e_j|f_{a_j})$ where the dependency on all previous words is dropped. The parameters of IBM1 are $\boldsymbol{\theta} = \{p(e|f), \forall (e, f) \in \Lambda \times \Sigma\}$, where source and target vocabularies are restricted to the words encountered in the training corpus. Note that this already corresponds to a large number of parameters $|\Lambda| \times |\Sigma|$.

[3]For the sake of clarity, we make the parameters explicit in the notation of the distribution $p_{\boldsymbol{\theta}^{(t)}}$.

Inference and EM Even though enumerating all possible alignments is intractable, the strong independence assumptions of IBM1 make inference very efficient. Since alignment decisions are independent from one another, the best alignment is found my maximizing the probability of each alignment link: $\forall i, a_i^* = \arg\max_{a_i \in \mathcal{J}} p(e_i|f_{a_i})$.

The simplicity of the model structure allows for efficient computation of the posteriors in the E-step of EM. The computational complexity of the summation over the alignment space can be reduced from $O(M^N)$ to quadratic $O(MN)$. Furthermore, Brown et al. (1993) show that the log-likelihood function is concave, which guarantees obtaining a global maximum with EM. In a recent paper, Toutanova and Galley (2011) show that IBM1 it is not strictly convex, and there is a large space of parameter values that achieve the same optimal value of the objective. They perform several experiments to study the achieved variance in parameters resulting from different random initialization in EM, and the impact of initialization on test set log-likelihood and alignment error rate. Their experiments suggest that initialization does matter in practice, contrary to the views of (Brown et al., 1993).

Limitations The only information that IBM1 uses is the word co-occurrence which makes it similar to the alignment heuristic presented in 2.2.1. The heuristic alignment methods of section 2.2.1 and the IBM1 model both have shortcomings related the bag-of-word assumption. There is no model of distortion, the information about word positions is discarded. There is no possible way to control the number of target words aligned to some source word. These problems cannot be remedied without significantly altering the structure of the model, as discussed later. However, two other limitations of IBM1 are not deeply structural and are addressed by Moore (2004a) by merely changing the parameter estimation. These non-structural problems are:

- **Garbage collectors.** Due to the maximization of the likelihood during EM, it is sometimes beneficial to align many words in the target to some rare source word. Such rare words act as "garbage collectors" (Brown et al., 1993; Och and Ney, 2004). This problem is not specific to IBM1 but it is worst that in other models because of its simple structure. Moore (2004a) suggests that smoothing lexical probability limits this effect.

- `null` **alignment**. Too few target words get aligned to the source `null` word. This is because the model has only one such token. Adding multiple `null` words improve the alignment. `null` alignments are useful to account for corpus quality and translation phenomena corresponding to deletion/insertion of words.

IBM Model 2 IBM2 extends the previous model with a *distortion* model, which introduces a dependency on the absolute position of the source word. This dependency encodes preference for some alignment patterns, helping, for instance, to select source positions that are close to the diagonal of the alignment matrix.

$$p(\mathbf{e}, \mathbf{a}|\mathbf{f}) = p(M|N) \prod_{j=1}^{M} p(a_j|j, N, M) p(e_j|f_{a_j}). \tag{2.10}$$

$p(a_j|e_1^{j-1}, a_1^{j-1}, f_1^N)$ is no longer uniform and is simplified as $p(a_j|j, N, M)$. The dependency on M is usually dropped to further reduce the number of parameters.

Similar to IBM1, the simple structure still allows for efficient computation of the summation over all possible alignments. Nevertheless, unlike IBM1, the likelihood objective is no longer concave and EM is guaranteed to converge only to a local maximum. The parameters obtained from training IBM1 are often used to initialize the lexical parameters of IBM2. This helps EM find a better point of the likelihood function.

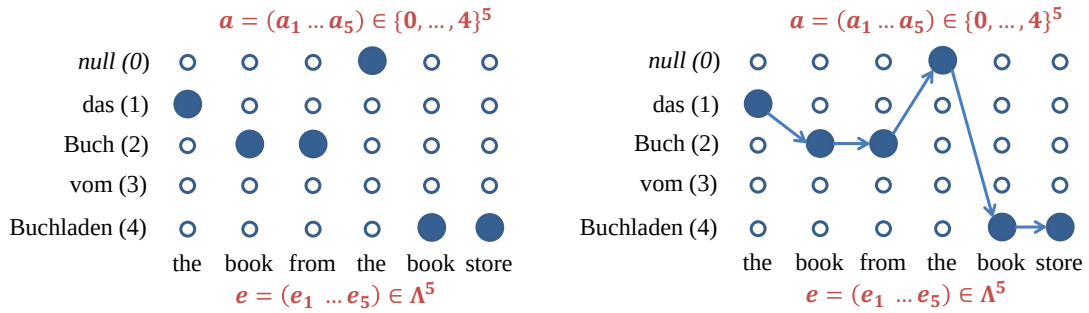


Figure 2.4: *Sequence labeling with bi-gram alignment dependencies.*

Hidden Markov Model (HMM) alignment Preference to monotonic alignment as reflected by the structure of IBM2 can be refined by modeling interactions between alignment decisions.

Translation is generally monotonic, hence the translations of two consecutive words in one language are probably placed near each other in the other language. Another example of dependency is linguistic patterns. When translating the Arabic pattern `verb noun` to English, word positions are inverted. This dependency can be captured by modeling the distortion of the translation, in this case $a_j - a_{j-1} = -1$.

Dagan, Church, and Gale (1993); Vogel, Ney, and Tillmann (1996) propose to model the alignment as first-order Hidden Markov Model (Baum and Petrie, 1966). The translation probability factors according to this model as:

$$p(\mathbf{e}, \mathbf{a}|\mathbf{f}) = p(M|N) \prod_{j=1}^{M} p(a_j|a_{j-1}, N) p(e_j|f_{a_j}). \tag{2.11}$$

The transition probability depends only on the jump width in (Vogel, Ney, and Tillmann, 1996): $p(a_j|a_{j-1}, N) = p(a_j - a_{j-1})$. Models parameters are then the emission (same as for IBM1) and transition probabilities. Figure 2.4 illustrates the added dependency. Several approaches to enhance the integration of the `null` word in the model have been proposed in the literature (Och and Ney, 2003; Liang, Taskar, and Klein, 2006).

The HMM model has attractive properties which make the basis for many extensions. Toutanova, Ilhan, and Manning (2002) propose several models based on the HMM to address different problems. These extensions aim to boost lexical translation probabilities with *part-of-speech (POS)* tags; better modeling of the `null` alignments; and incorporating the notion of *fertility* that we explain in the following IBM models.

Inference and EM The first-order HMM encodes dependencies between consecutive labels and satisfies the optimal substructure requirement for dynamic programming. The best alignment sequence can therefore be found using the Viterbi algorithm (Viterbi, 1967) with computational complexity quadratic in the sequence length.

Parameter estimation is similar in principle to the IBM models. However, the log-likelihood function in HMM models is not concave and EM is capable of finding only a local maximum. EM is therefore initialized with the parameters of a trained IBM1 model to ensure a good starting point. Summing over all possible alignment sequences in order to compute the posterior probabilities can be done efficiently using the Baum-Welch algorithm (Baum et al., 1970). Once the posteriors are computed, count expectations are accumulated over the entire training corpus and the M-step is performed as before.

IBM model 3 Languages express meaning using different number of words per concept. The English word "`potatoes`" for example, translates to "`pommes de terre`". Hence the

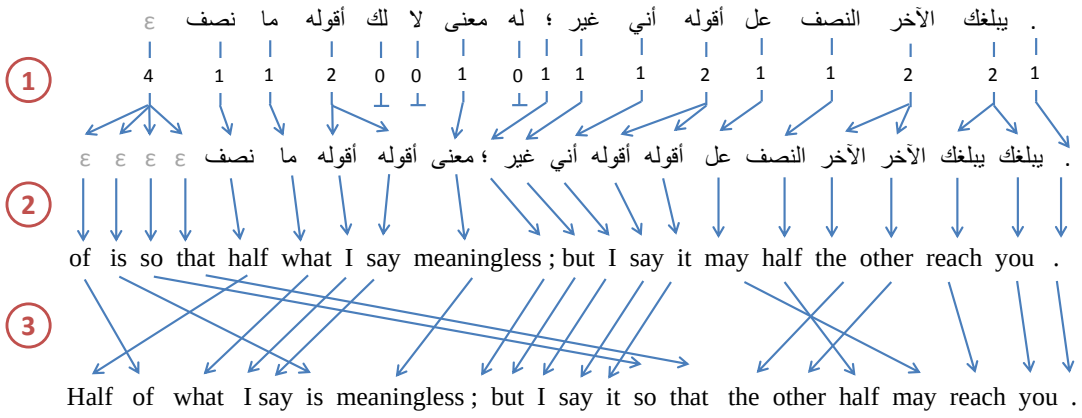


Figure 2.5: *Generative story: starts with a fertility step (1), followed by a lexical substitution step (2) and ends with a distortion step (3).*

tendency of some source words to align with more target words than others is an important phenomena. Yet, **word fertility** (Brown et al., 1993) is not accounted for in the models presented previously. Each source word is said to have a fertility $\phi = 0, 1, 2, ...$ equals to the number of corresponding target words. So in the previous example, the alignment model has no explicit preference to align the three French words "`pommes de terre`" to the same English word "`potatoes`". It leaves the decision entirely to lexical probabilities.

With IBM3, Brown et al. (1993) propose to enrich the IBM model 2 by adding for each source word f_i, a probability distribution over possible fertilities $p(\phi|f_i)$. One the one hand, if the distribution $p(\phi|\texttt{potatoes})$ is peaked at $\phi = 3$, the model will assign a higher probability to alignment containing three links involving "`potatoes`". Similarly, the tendency of some source words, such as the English auxiliary `do`, to remain unaligned can be reinforced. On the other hand, the number of target words to be aligned with the source `null` can be controlled by setting its fertility. Brown et al. (1993) define the `null` fertility distribution $p(\phi_0)$ as a function of the sentence target length and a parameter p_0 representing the *a priori* probability of a `null` alignment.

In IBM3, like in all generative models, the probability $p(\mathbf{e}, \mathbf{a}|\mathbf{f})$ factorizes according to the model generative story. A pictorial example of the generative story of IBM3 is represented in Figure 2.5. Each step in the model admits several alternatives, each of which is associated with a parameter in generative modeling. Computing the probability of a given structure amounts to multiplying the parameters as prescribed by the generative story. In Figure 2.5 one of the parameters is $p(\phi = 2|\text{أقوله})$. A useful way to enumerate all possible structures and to compute their probabilities is to use a *cascade* of finite-state transducers (FSTs) (Mohri, 1997) as described in (Knight and Al-Onaizan, 1998).

Inference and EM Modeling fertility comes at the price of an increased complexity. A new set of $\mathbf{F}_{\text{max}}$ parameters are needed for each source word $f \in \Sigma$ to represent the fertility distribution. More importantly, due to the added dependencies, the search for the most probable alignment under IBM3 is NP-hard (Udupa and Maji, 2006) and can not be performed exactly. One technique for finding good solutions is to use alignments produced by IBM2 as a starting point for heuristic *hill-climbing* techniques, and to explore their neighboring alignments by applying local modifications on the alignment. Other techniques are possible as well (Brown et al., 1993; Och and Ney, 2003; Koehn, 2010). Such heuristics are also used to *sample* the search space and to construct a set of high-probability alignments used as an approximation of the search space which is used by Expectation-Maximization (EM) to

compute the required statistics.

IBM model 4 and beyond Although IBM3 already covers many essential properties of alignments, it still makes a lot of assumptions. Its parameters are still independent of surrounding contexts and interactions between alignment decisions are not explicitly considered.

IBM model 4 brings several improvements to IBM3:

- Distortion is modeled with relative positions instead of absolute ones, which helps achieve a better generalization and reduce the effect of data sparsity.
- A first-order dependency between alignment decisions is introduced, which captures the tendency of *chunks* of words to move together.
- A dependency on word classes for distortion models, which incorporate lexical knowledge while dealing with data sparsity. Word classes are computed automatically in an unsupervised way (Brown et al., 1993).

As with IBM model 3, training this model is very expensive and exhaustive count collection is impossible. Hill-climbing techniques, based on model 3 alignments, are used in the same manner as training is performed for IBM3.

Fertility-based models 3 and 4 are **deficient** in the sense that they waste probability mass on impossible alignment structures. This is because they ignore whether or not a source position has been chosen; and probability mass is reserved for source positions outside the sentence boundaries (Brown et al., 1993; Och and Ney, 2003). A fix is proposed in IBM model 5 Brown et al. (1993) at the expense of additional training complexity and of an increase in the number of parameters. Such additional complexity is not accompanied with visible gains in model performance and hence IBM4 is more used in practice.

Local log-linear parameterization All the models described so far use a multinomial-based parameterization of the CPDs. A Log-linear parameterization can be applied in two ways. The first is to define a single globally normalized log-linear model (Markov field) for the joint distribution i.e. over the entire space $\Lambda^* \times \mathcal{A}$:

$$p(\mathbf{e}, \mathbf{a}|\mathbf{f}) = \frac{\exp \boldsymbol{\theta}^\top \mathbf{g}(\mathbf{e}, \mathbf{a}, \mathbf{f})}{Z(\boldsymbol{\theta}, \mathbf{f})}. \tag{2.12}$$

The resulting partition function ($Z(\boldsymbol{\theta}, \mathbf{f})$) must sum over a very large space, and approximations are often required.

The second way to use a log-linear parameterization in the generative setting is to use log-linear distributions over derivation steps in the generative process. In this view, Berg-Kirkpatrick et al. (2010) propose to re-parameterize the emission model in IBM1 and HMM with a log-linear model instead of a multinomial. The motivation for this parameterization is two-fold:

- It enables to use hand-designed features to declaratively integrate domain knowledge into a model without having to worry about their dependencies. An example of such feature would be testing whether the source and target words are both capitalized.
- Simple training in the unsupervised setting due to the locality of the feature functions. Optimizing the likelihood objective does not require to compute expectations over the joint distribution as in globally normalized Markov field. EM can still be applied with the E-step unchanged, and the M-step involving standard gradient-based optimization.

Berg-Kirkpatrick et al. (2010) obtain improvement in performance over IBM1 and HMM by using simple features functions of the involved words, including: edit distance, stem, prefixes, appearance in a dictionary, etc. The same log-linear parameterization of the HMM has been proposed in (Varea et al., 2001) but it was trained using supervised estimation techniques. In fact, using rich features in log-linear parameterization is more widely used in globally normalized conditional models, trained with supervised methods (See Section 2.2.3).

Discussion Word-based unsupervised generative models are widely used in practice since they only require a sentence-aligned parallel corpus to train. However, they have several drawbacks. Incorporating additional features is not straightforward. Additionally, large amount of training data is required in order to obtain reasonable results.

2.2.2.2 Conditional Random Fields

The models previously described for lexical distribution are locally normalized. While the CPDs of the joint models may be parameterized with log-linear models (Berg-Kirkpatrick et al., 2010) (cf. 2.2.2.1), the requirement that models factorize according to a particular generative process imposes a considerable restriction on the kinds of features that can be incorporated[4].

Instead of locally normalized models, we will now describe a *globally normalized log-linear* model, also called a CRF (Lafferty, McCallum, and Pereira, 2001). The idea is to score each input-output pair with a linear score which is normalized to a well-formed probability:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp \boldsymbol{\theta}^\top \mathbf{g}(\mathbf{x}, \mathbf{y})}{\sum_{\acute{\mathbf{y}} \in \mathbf{Y}_\mathbf{x}} \exp \boldsymbol{\theta}^\top \mathbf{g}(\mathbf{x}, \acute{\mathbf{y}})} \tag{2.13}$$

The independence assumptions made by such models are usually represented using a an undirected graph called a *Markov Network*. The linear score factors according to the network structure into local parts called *clique potentials*. The structure of these cliques is important for efficient exact decoding. However, the probability does not necessarily factor according to derivation steps or to a generative process.

Dyer et al. (2011) use such a model for the distribution $p(\mathbf{e}, \mathbf{a}|\mathbf{f})$ over the target sentence and the hidden alignment variable. It can incorporate arbitrary, overlapping features, and it can be used to infer word alignments[5]:

$$p(\mathbf{e}, \mathbf{a}|M, \mathbf{f}) = \frac{1}{Z(\boldsymbol{\theta}, \mathbf{f})} \exp \sum_{j=1}^{M} \sum_{h=1}^{|\boldsymbol{\theta}|} \theta_h g_h(a_j, a_{j-1}, e_j, e_{j-1}, M, \mathbf{f}). \tag{2.14}$$

The model enjoys the usual benefits of discriminative modeling, but is trained entirely from parallel sentences without gold-standard word alignments. Figure 2.6, borrowed from (Dyer et al., 2011) compares the CRF and the CBN structure for the IBM models.

For a given source sentence $\mathbf{f} \in \Sigma^*$, the model defines a distribution over all possible translations $\mathbf{e} \in \Lambda^*$ and all possible alignments that can be built for $(\mathbf{e}, \mathbf{f})$. The feature functions used in this model perform many tests including word association measures, positional information, lexical features similar to previous models, Hidden Markov Model (HMM)-like path features, etc. The families of features used in the literature will be discussed in Section 2.6.

[4] We refer the reader to (Koller and Friedman, 2009) for a full discussion of these alternative representations.

[5] Blunsom and Cohn (2006); Allauzen and Wisniewski (2010) describes a similar model which encodes the distribution $p(\mathbf{a}|\mathbf{e}, \mathbf{f})$ directly (cf. Section 2.2.3)

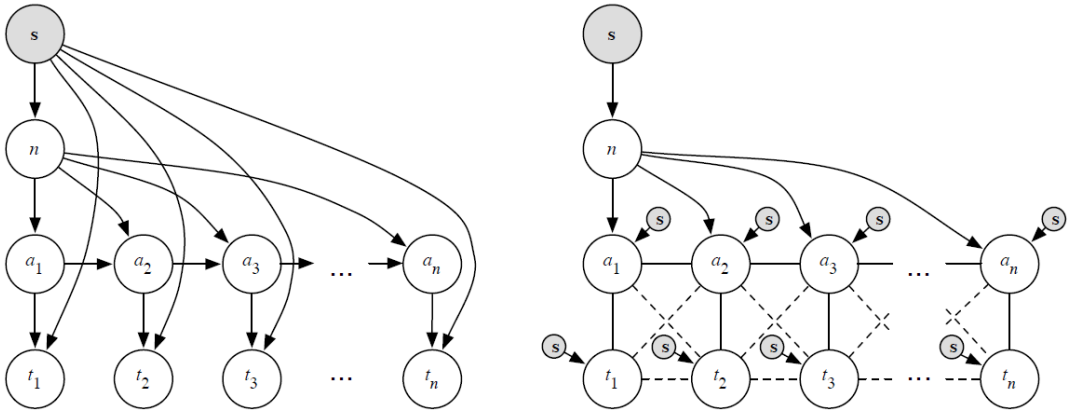


Figure 2.6: *On left is the conditional Bayesian network that encode dependencies in locally normalized models (Brown et al., 1993; Berg-Kirkpatrick et al., 2010). On the right is the conditional random field used by (Dyer et al., 2011) from which the figure is adopted.*

Inference The more dependencies the structure encodes, the harder exact inference is. As can be seen from Equation (2.14), Dyer et al. (2011) design their features so as to keep the width of the tree-decomposition of the graphical model sufficiently low to allow exact inference. Under their independence assumptions, exact inference is tractable using dynamic programming.

Unsupervised parameter estimation In the absence of alignment annotations, the parameters $\boldsymbol{\theta}^*$ are selected to maximize the *marginal conditional log-likelihood*:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{\tilde{N}} \log \sum_{\mathbf{a} \in \mathcal{A}} p_{\boldsymbol{\theta}}(\mathbf{e}, \mathbf{a}|\mathbf{f}). \tag{2.15}$$

This objective is usually augmented with a *regularization* term in order to avoid overfitting which leads to Maximum *a posteriori* Estimation (MAP estimation) of the parameters. Norms of the parameter vector, such as $\ell_1 = \|\boldsymbol{\theta}\|_1$ or $\ell_2 = \|\boldsymbol{\theta}\|_2^2$ or a combination thereof, are widely used in practice. Regularization strength can be tuned to optimize some quality measure, AER for instance.

Due to the presence of a hidden variable, the above objective is non-convex in the model parameters $\boldsymbol{\theta}$. Therefore, algorithms that find a local optimum have to be used. Dyer et al. (2011) use an online method that approximates ℓ_1 regularization and only depends on the gradient of the unregularized objective (Tsuruoka, Tsujii, and Ananiadou, 2009).

2.2.3 Supervised Discriminative Sequence Models

Models presented in Section 2.2.2 define the joint probability distribution $p(\mathbf{e}, \mathbf{a}|\mathbf{f})$ and use it to infer the alignment. They are all trained in an unsupervised way. In this section we consider supervised models for sequence structure prediction.

2.2.3.1 Maximum entropy models

The simplest approach is to directly estimate, for each target word, the probability the alternative alignment decisions which range over the source positions. This can be done using

a popular multi-class classification framework called MaxEnt, of which CRF is a generalization to more complex structures.

Ittycheriah and Roukos (2005) propose to model the conditional alignment distribution using a log-linear model:

$$p(\mathbf{a}|\mathbf{e},\mathbf{f}) = \frac{1}{Z(\alpha,\mathbf{e},\mathbf{f})} \exp \sum_{j=1}^{M} \alpha p(a_j|a_{j-1}) + (1-\alpha) p(a_j|e_1^{j-1},\mathbf{f}). \tag{2.16}$$

The second term $p(a_j|e_1^{j-1},\mathbf{f})$ is an observation model which measures the strength of association between a source word and a target word, using a set of feature functions extracted from the words and their context. The parameters of this model learned from an annotated parallel corpus. Inference in the observation model is performed in polynomial time since each word is labeled separately. The first term $p(a_j|a_{j-1})$ is a transition model, in which each alignment link depends on the previous one. Therefore, *Beam search* is used to find the alignment that maximizes the overall model $p(\mathbf{a}|\mathbf{e},\mathbf{f})$. One problem of this model is that in order to take advantage of the transition model, a large beam must be maintained, which slows down the inference. Additionally, the parameter α is fixed to 0.5 by hand and not learned from data.

The next model combine both the transition and the observation into a single model in straightforward way using a CRF.

2.2.3.2 Conditional Random Fields

Blunsom and Cohn (2006) describe a discriminative sequence labeling model that directly encodes the distribution $p(\mathbf{a}|\mathbf{e},\mathbf{f})$ using a linear-chain CRF. With a structure similar to a HMM exact inference and efficient learning algorithms are available through adaptations of the Viterbi and forward-backward algorithms (Sutton and Mccallum, 2007). The model is given as[6]:

$$p(\mathbf{a}|\mathbf{e},\mathbf{f}) = \frac{1}{Z(\boldsymbol{\theta},\mathbf{e},\mathbf{f})} \exp \sum_{j=1}^{M} \sum_{h=1}^{|\boldsymbol{\theta}|} \theta_h g_h(a_j, a_{j-1}, \mathbf{e}, \mathbf{f}). \tag{2.17}$$

The output variable of this model is significantly less complex than the model described in Section 2.2.2.2: for a given sentence pair from the input space $(\mathbf{e},\mathbf{f}) \in \Sigma^* \times \Lambda^*$ the model defines a distribution over all possible alignments that can be built for $(\mathbf{e},\mathbf{f})$ under the constraints of the model. Conditioning on both sentences allows for wider range of cheap features than in the model in (Dyer et al., 2011). However, this requires the availability of alignment information during training as we will see next. Again, discussion of feature functions choice is differed to Section 2.6. The best alignment is found using the Viterbi algorithm, similar to inference with HMMs.

Supervised parameter estimation As with generative models, the parameters can be selected to maximize the *conditional log-likelihood*. Since the only modeled variable is the alignment, MLE requires a corpus annotated with alignment information for training $\{(\tilde{\mathbf{e}}_k, \tilde{\mathbf{f}}_k, \tilde{\mathbf{a}}_k)\}_{k=1}^{\tilde{N}}$.

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \frac{1}{\tilde{N}} \sum_{k=1}^{\tilde{N}} \boldsymbol{\theta}^\top \mathbf{g}(\tilde{\mathbf{e}}_k, \tilde{\mathbf{f}}_k, \tilde{\mathbf{A}}_k) - \log Z(\boldsymbol{\theta},\mathbf{e},\mathbf{f}). \tag{2.18}$$

[6] A artificial start token is added to the sentence at position a_0 since the index of a sentence starts at 1.

Unlike joint models based on multinomial distributions, MLE for conditional log-linear models does not have a closed-form solution. This is the price to be paid for allowing arbitrary features. However, Equation 2.18 defines an unconstrained optimization problem of a function that is smooth, differentiable and globally concave. Its global maximum can be obtained using numerical optimization methods such as L-BFGS (L-BFGS) (Liu and Nocedal, 1989).

Blunsom and Cohn (2006) perform MAP estimation of the parameters, which can be done efficiently for their model: the partition function and expected feature values can be computed efficiently with DP. Instead of ℓ_1 regularization, Blunsom and Cohn (2006) include a Gaussian prior over the parameters[7].

2.2.3.3 Large-Margin methods

Similarly to the MaxEnt model described in Section 2.2.3.1, Ma et al. (2008) propose to label each target word with the source position having the maximal association score, where the score is computed as:

$$score(e_i, f_j) = \boldsymbol{\theta}^\top \mathbf{g}(\mathbf{e}, \mathbf{f}, \mathbf{A}). \tag{2.19}$$

This multi-class classification problem is solved using Support Vector Machines (SVM) (Cristianini and Shawe-Taylor, 2000), incorporating predictions of generative alignment models (like IBM and HMM) as features, in addition to various syntactic and linguistic features.

In linear methods, such as perceptron and large-margin algorithms, the goal is to learn a function which score each input-output pair with a linear combination of features functions.

Training considers the point $\mathbf{g}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ and all competing points $\mathbf{g}(\tilde{\mathbf{x}}_i, \mathbf{y})$ for $\mathbf{y} \neq \tilde{\mathbf{y}}_i$. The goal is to choose a direction (encoded in the weight vector $\boldsymbol{\theta}$) along which the point $\mathbf{g}(\mathbf{x}_i, \mathbf{y}_i)$ has a high score. Furthermore, the alternative points $\mathbf{g}(\mathbf{x}_i, \mathbf{y})$ should all receive scores that are inversely proportional to the amount of error incurred in labeling $\tilde{\mathbf{x}}_i$ with $\mathbf{y}$ when the true answer is $\tilde{\mathbf{y}}_i$. This is naturally encoded in the cost function $cost(\tilde{\mathbf{x}}_i, \mathbf{y}, \tilde{\mathbf{y}}_i; h)$, which now becomes an abstract component of the learner. For a detailed discussion on large margin methods we refer the reader to Taskar (2004).

Despite this different interpretation, the SVM cost-augmented objective is very similar to the ℓ_2-regularized maximum *a posteriori* objective of the previous model (Gimpel and Smith, 2010). However, large margin methods are purely discriminative: they aim to perform well on the task defined by their cost function. In other words, if we know how a model is to be evaluated at decoding time, a cost function can be defined for use at training time, providing an opportunity to better inform the learner about its real goals.

2.3 Symmetric Many-to-Many Methods

Sequence labeling approaches studied in Section 2.2 produce asymmetric one-to-many alignments. However, the one-to-many assumption is over-simplistic and relies on an arbitrary choice of the alignment direction[8].

A different approach that does not suffer from asymmetry is to predict the *binary alignment matrix*. The problem is reformulated as follows:

- Input is a pair of sentences $(\mathbf{e}, \mathbf{f}) \in \Sigma^* \times \Lambda^*$.
- The output is an unrestricted many-to-many word alignment $\mathbf{A} \in \mathcal{A}$. This alignment is usually represented by enumerating the functions $A : \mathcal{N} \times \mathcal{M} \rightarrow \{0, 1\}$ which map the cells (i, j) in the *alignment matrix* to a binary value $A_{i,j}$ indicating whether corresponding words are aligned or not: where 1 indicates an *active* link and 0 an *inactive* link.

[7] Zero-mean Gaussian prior with uniform covariance matrix is equivalent to the ℓ_2 regularization (Chen and Goodman, 1996).

[8] Unless it is used for translation.

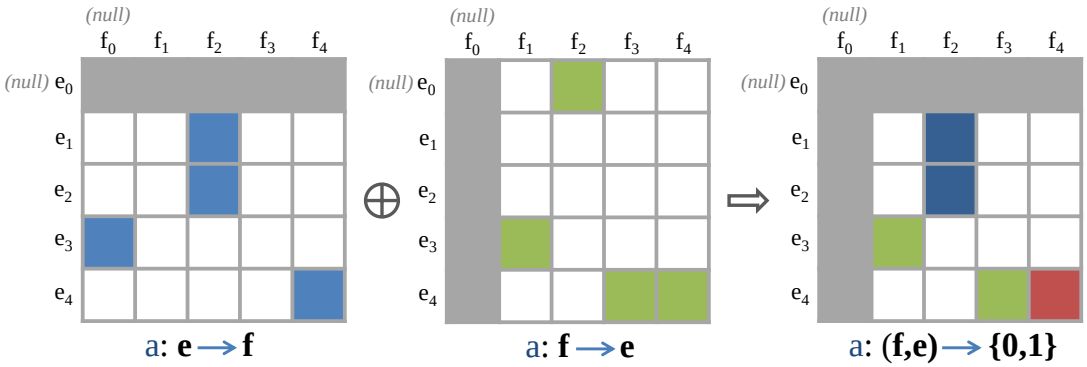


Figure 2.7: *Combination of two directional alignments. Colored points represent the union while the red point represents the intersection.* `null` *alignment are ignored.*

2.3.1 Symmetrization and Alignment Combination

An increase in alignment model's expressivity usually comes at the price of intractability, implying approximate heuristic learning and inference prone to search errors. Going beyond IBM model 2 or HMM is an example. An alternative is to combine several simple alignments to obtain a more expressive one.

2.3.1.1 Symmetrization heuristics

The simplest approach is to merge the two directional alignment functions using a *symmetrization heuristic* (Och, Tillmann, and Ney, 1999; Koehn, Och, and Marcu, 2003; Och and Ney, 2003).

One such heuristic is to take the *intersection* of the two alignment sets as follows[9]: $\mathbf{A} = \mathbf{a}_{\mathbf{f}\to\mathbf{e}} \cap \mathbf{a}_{\mathbf{e}\to\mathbf{f}}$. Intersection alignments matrices are sparse and encode only one-to-one relationship between words. However the alignment are usually of high precision due to the agreement of both models.

An alternative assumption is that the two alignments contain complementary information and their *union* is therefore considered instead of their intersection. Many-to-many relationship can be expressed this time and the resulting matrices tend to be highly populated. A higher recall can be achieved at the price of losing in precision.

Figure 2.7 depicts the space of possible links considered by the heuristic.

There exists any number of mid-ground solutions which aim to balance precision and recall. One could start from high precision intersection points, and gradually add reliable links from the union to increase recall, or go the other way around, starting from the high recall union points and remove unreliable links to increase precision. *Growing* heuristics which iteratively add links from the *neighborhood* of reliable links until no word is left unaligned, generally achieve good performance. The most famous heuristic in this family is called *grow-diag-final-and* (Koehn, Och, and Marcu, 2003).

Grow-diag-final-and (GDFA) GDFA is a simple heuristic which performs very well in practice, and is widely used in state of the art translation systems. We use this heuristic in Part II as one of the baselines to which we compare our models.

GDFA starts from the intersection of two directional alignments. The "grow-diag" step considers the neighborhood $\{(\acute{\imath}, \acute{\jmath})\}$ of each point (i, j) in the intersection, where the neighborhood contains the points, the source index of which is in the range $[i-1, i+1]$ and the

[9] Equivalently, $\mathbf{A}_{i,j} = 1 \iff (i,j) \in \mathbf{a}_{\mathbf{f}\to\mathbf{e}} \cap \mathbf{a}_{\mathbf{e}\to\mathbf{f}}$, and $\mathbf{A}_{i,j} = 0$ otherwise.

target index is in the range $[j-1, j+1]$. Points in this neighborhood are progressively added to the alignment if neither the source word nor the target word is already aligned and the corresponding point exists in the union $\mathbf{a}_{\mathbf{f}\to\mathbf{e}} \cup \mathbf{a}_{\mathbf{e}\to\mathbf{f}}$. At the end, the "final" step aligns whatever source and target words that remained unaligned if an appropriate point exists in the union.

Generalizing the symmetrization While the main reason for using such heuristic is to symmetrize the 1-to-many alignments, they can be easily generalized to more than two alignments $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ that are not necessarily asymmetric. This is done simply by taking their union $\bigcup_{k=1}^{n} \mathbf{a}_k$ (the intersection is analogous)[10].

Additional clues can be encoded to the heuristic constraints. External linguistic knowledge for instance is incorporated by (Crego and Habash, 2008) based on the intuition that words inside the same *chunk* in one language tend to align to words inside one *chunk* in the other.

Application-driven combination Deng and Zhou (2009) perform combination in light of an intended application of the resulting alignments. Like the heuristic, the aim is to find a balance between intersection and union. But unlike the heuristics, combination is carried out as an optimization process driven by an effectiveness function. This function evaluates the impact of each alignment link on the number of phrase pairs that can be extracted from the sentence pair[11]. Thus, the word alignment combination is seen as a process of maximizing the number of extracted phrase pairs.

2.3.1.2 Agreement constraints

Instead of symmetrizing the "Viterbi" output of directional models *a posteriori*, one can jointly maximize a combination of data likelihood and agreement between the models.

Viewing intersection as a way of finding predictions that both models agree on, Liang, Taskar, and Klein (2006) modify the objective to incorporate both data likelihood and a measure of agreement between models, which is quantified using the probability that the alignments produced by the two models ($p_{\theta_1}(\mathbf{e}, \mathbf{a}|\mathbf{f})$ and $p_{\theta_2}(\mathbf{f}, \mathbf{a}|\mathbf{e})$), agree on an example $\mathbf{x} = (\mathbf{e}, \mathbf{f})$. The objective function used for training becomes[12]:

$$\max_{\theta_1,\theta_2} \sum_{(\mathbf{x})} \left[\log p_{\theta_1}(\mathbf{x}) + \log p_{\theta_2}(\mathbf{x}) + \log \sum_{\mathbf{a}} p_{\theta_1}(\mathbf{a}|\mathbf{x}) p_{\theta_2}(\mathbf{a}|\mathbf{x}) \right]. \tag{2.20}$$

However the product distribution $p_{\theta_1}(\mathbf{a}|\mathbf{x})p_{\theta_2}(\mathbf{a}|\mathbf{x})$ ranges over all one-to-one alignments and computing it is #P-complete (Valiant, 1979; Liang, Taskar, and Klein, 2006).

A variety of approximate probabilistic inference techniques, for example, sampling or variational methods can be used. In practice, a simple approximation that uses posterior marginal probability of individual links $p_{\theta}(a_{i,j}|\mathbf{e}, \mathbf{f})$ works well. Such probabilities, which are called *state occupation probabilities* in (Matusov, Zens, and Ney, 2004) are computed efficiently for simple models (Baum-Welch for HMM).

One problem in this procedure is that it is not clear what objective the approximate procedure actually optimizes. Ganchev, Graca, and Taskar (2008); Graça, Ganchev, and Taskar (2010) incorporate agreement constraints to EM training using Posterior Regularization (PR) (Graça, Ganchev, and Taskar, 2007) aims to incorporate side-information into unsupervised estimation in the form of constraints on the model's posteriors. Such constraints are useful

[10] Note that the combined alignments need not be directional; any alignment $\mathbf{A}$ can be used.

[11] See Section 2.5.2.1 for phrase pairs extraction methods.

[12] The distributions $p_{\theta_1}(\mathbf{x}) = p(\mathbf{e})p_{\theta_1}(\mathbf{f}, \mathbf{a}|\mathbf{e})$ and $p_{\theta_2}(\mathbf{x}) = p(\mathbf{f})p_{\theta_2}(\mathbf{e}, \mathbf{a}|\mathbf{f})$ are used in the equation in order to unify the notation and remove the condition. Since both $\mathbf{e}$ and $\mathbf{f}$ are known in each respective model, $p(\mathbf{e})$ and $\mathbf{p}(\mathbf{f})$ do not affect the training.

for several reasons. As with any unsupervised induction method, there is no guarantee that the maximum likelihood parameters correspond to the intended meaning for the hidden variables; and constraining the expected value of some features instead of adding them to the generative story of the model enables to express features that would otherwise make the model intractable.

For example, enforcing that each hidden state of an HMM should be used at most once per sentence would break the Markov property and make the model intractable. In contrast, using the PR framework, one can enforce the constraint that each hidden state is used at most once in expectation. The underlying model remains unchanged, but the learning method changes. During learning, the method is similar to the EM algorithm with the addition of solving an optimization problem similar to a maximum entropy problem inside the E-Step. Graça, Ganchev, and Taskar (2010) shows how to add *Bijectivity* and *Symmetry* constraints. We use an implementation of this model called Posterior Constrained Alignment Toolkit (PostCAT)[13] (Graça, Ganchev, and Taskar, 2007; Ganchev, Graca, and Taskar, 2008; Graça, Ganchev, and Taskar, 2010) as a baseline in our experiments in Part II.

Once the model parameters are trained, the output alignment can be obtained either using the Viterbi algorithm or using Minimum Bayes-Risk (MBR) decoding (Kumar and Byrne, 2004) as discussed in Section 2.3.2.1.

Instead of training two separate models, DeNero and Macherey (2011) propose to embed two directional HMM aligners into a single model. While the combined model structure rewards agreement, the inference is intractable due to numerous cycles in the model's graph. Dual decomposition (Sontag, Globerson, and Jaakkola, 2011) is used as an approximate inference technique.

2.3.1.3 Discriminative combination

Instead of making combination decisions heuristically, or modifying the generative training procedure, one would wish to combine several clues in a more principled way. The discriminative modeling framework offers the possibility to combine feature functions while optimizing a well-defined objective. A binary classifier can then be used to compute the function $A_{i,j} \in \{0,1\},\ \forall (i,j) \in \bigcup_{k=1}^{n} \mathbf{a}_k$.

Several such models have been investigated in the literature. For instance Ayan and Dorr (2006a) propose to use an MaxEnt classifier to combine all IBM and HMM models. Learning is performed in a supervised way to maximize the regularized conditional likelihood of manual word alignments for a small parallel corpus. Features include the generative alignment predictions and external linguistic information such as Part-of-Speech (PoS) tags of source and target words. Elming and Habash (2007) combine alignments obtained from several preprocessing (tokenization) schemes using a rule-based classifier. Ma et al. (2008) start from the intersection of IBM models (or a heuristic) to build a high precision *anchor* set used as features in a SVM classifier. Fossum, Knight, and Abney (2008) use greedy search algorithms with a linear scoring function to decide which links should be removed from the union. The parameter of this scoring function are estimated using the averaged perceptron algorithm with structured outputs (Collins, 2002). However, this scoring function is defined globally at the level of the entire alignment structure and not at the level of individual links. We will discuss such methods more in details in Section 2.3.4.

2.3.2 Weighted Matrix Based Methods

Several alignment methods associate a score $c_{i,j}$ to each link (i,j) in the alignment matrix and search for the alignment $\mathbf{A}$ with the maximal score under some constraints. We refer to

[13] http://www.seas.upenn.edu/~strctlrn/CAT/CAT.html

the cost matrix as the Weighted Alignment Matrix (WAM):

$$\mathbf{C} = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,M} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N,1} & c_{N,2} & \cdots & c_{N,M} \end{pmatrix} \tag{2.21}$$

Several approaches to compute the individual scores exist, among them:

- Corpus statistics such as Pearson's χ^2 (Gale and Church, 1991) or the likelihood ratio (Dunning, 1993), and Melamed (2000) probabilistic noise model;
- Context information can be incorporated in a probabilistic model using several feature functions (Cherry and Lin, 2003);
- Link posterior probabilities under some alignment model or a combination thereof (Matusov, Zens, and Ney, 2004; Liang, Taskar, and Klein, 2006; DeNero and Klein, 2007; Graça, Ganchev, and Taskar, 2010);
- Weighted linear combination of multiple feature scores (Tiedemann, 2003b; Taskar, Lacoste-Julien, and Klein, 2005; Ren, Wu, and Wang, 2007).

Once the cost matrix is built several types of constraints and search algorithms can be applied, even including image processing techniques (Chang and Chen, 1997b). In the following we review the most widely used approaches.

2.3.2.1 Minimum Bayes-risk decoding

Under probabilistic models, the output alignment is normally predicted by selecting the single best (Viterbi) alignment given the model parameters.

Another possibility is to use Minimum Bayes-Risk decoding (Kumar and Byrne, 2002; Liang, Taskar, and Klein, 2006; Graça, Ganchev, and Taskar, 2010), which uses posterior-based computed matrices. The alignment inference procedure includes a link if its score is above some threshold. The same method can be used with different type of matrices (Ren, Wu, and Wang, 2007). This allows the accumulation of probability from several low-scoring alignments that agree on one alignment link. The threshold is tuned on some small amount of labeled data to minimize some loss. MBR decoding has several advantages over the maximum probability decoding. First, irrespectively of the particular choice of the loss function, the threshold enables to trade-off precision and recall. Second, with this method, it is possible to ignore the null word probabilities which tend to be poorly estimated.

MBR decoding results in *many-to-many* alignments even though the underlying models use have different constraints.

2.3.2.2 One-to-many constraints

We have already described a heuristic approach which uses an association scores matrix for sequence labeling (cf. Section 2.2.1) and which results in *one-to-many* alignments. Matusov, Zens, and Ney (2004) use the same approach for HMM posterior matrices.

2.3.2.3 One-to-one constraints

Simple thresholding can lead to wrong alignments because of spurious relations may be discovered in the matrix due to the garbage collector effect (Moore, 2004a). Therefore, additional *one-to-one* constraints may be helpful (Melamed, 2000; Cherry and Lin, 2003; Tiedemann,

2003b; Matusov, Zens, and Ney, 2004). Melamed (2000) presents the *competitive linking algorithm*, which uses a matrix of association scores. First, the highest-ranking word position (i, j) is aligned. Then, the corresponding row and column are removed from the association score matrix. This procedure is iteratively repeated until every source or target language word is aligned. Matusov, Zens, and Ney (2004); Tiedemann (2004) find one-to-one alignment by applying the Hungarian algorithm to solve the maximum-weight bipartite matching problem.

2.3.2.4 Alignment as assignment

Taskar, Lacoste-Julien, and Klein (2005) use the same formulation as the maximum weighted matching problem for bipartite graphs. However, each individual score $c_{i,j}$ is modeled as a weighted feature vector using an arbitrary number of real-valued or binary feature functions:

$$c_{i,j} = \sum_{h=1}^{|\boldsymbol{\theta}|} \theta_h g_h(i, j, \mathbf{e}, \mathbf{f}). \tag{2.22}$$

The weight vector $\boldsymbol{\theta}$ is trained to minimize a prediction error on the training data. Taskar, Lacoste-Julien, and Klein (2005) use weighted Hamming distance for the loss and formulate a *large-margin* learning problem.

Lacoste-Julien et al. (2006) noted that this approach is limited by the restriction that words have fertility of at most one; and more importantly, first order correlations between consecutive words are modeled only indirectly through the one-to-one constraints. They, therefore introduce a parameterized model that penalizes different levels of fertility without increasing in computational complexity, and incorporates first-order interactions between alignments of consecutive words by formulating the alignment problem as a quadratic assignment problem. In addition to scoring individual links, they also define scores of pairs of links that connect consecutive words in an alignment.

2.3.2.5 Alignment as matrix factorization

Goutte, Yamada, and Gaussier (2004) show that rephrasing the alignment problem as orthogonal non-negative matrix factorization allows to obtain *many-to-many* alignments that respect two constraints: *coverage*, where all words must be aligned (`null` included) and *transitive closure*, meaning that if f_{i_1} is aligned to e_{j_1} and e_{j_2}, then any word f_{i_2} aligned to e_{j_1} must also be aligned to e_{j_2}. They also give an algorithm that solves this problem.

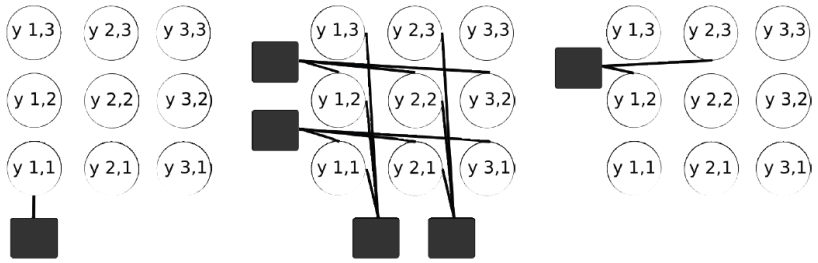
In a similar view, Deng and Gao (2007) use Singular Value Decomposition (SVD) as a prior knowledge to guide the alignment.

2.3.3 Generative Many-to-Many Models

Fraser and Marcu (2007a) describe a generative model called LEAF, which directly models many-to-many word alignments with gaps. This is different from the previous models, such as HMM, which authorize only one-to-many alignments. Reordering and fertility models in LEAF are similar to IBM model 4's. However, its nine-step generative story is considerably more complicated.

2.3.4 Global Discriminative Models

Models described here use a discriminative function to score entire alignment matrix structures using arbitrary global features and use a search guided by this score to make predictions. Compromises are needed when training the parameters because of the global features involved.



(a) Local features (b) Fertility features (c) First order features

Figure 2.8: *Dependencies and corresponding feature scopes in the CRF-based matrix model described (Niehues and Vogel, 2008).*

2.3.4.1 CRF-based matrix modeling

Niehues and Vogel (2008) model explicitly the dependencies in the alignment matrix into a CRF. We use this model as a discriminative baseline for our MaxEnt model presented in Chapter 4.

CRFs are most widely used for sequential structure prediction because exact inference is tractable. However, to model the alignment matrix, the graphical structure of the model needs to integrate more complex dependencies. The alignment matrix is described by a random variable $y_{i,j}$ for every source and target word pair (f_i, e_j). These variables can have two values, 0 or 1, indicating whether the corresponding words are aligned or not. A word with zero fertility is indirectly modeled by setting all associated random variables to a value of 0. The structure of the CRF is described by a factored graph which contains two different types of nodes: hidden nodes, which correspond to the random variables; and factored nodes shown in Figure 2.8, taken from (Niehues and Vogel, 2008). Local features along with global fertility and first-order features make the dependencies quite complex and subsume many loops in the graphical structure, so the loopy belief propagation algorithm is used for approximate inference. Our MaxEnt matrix, described in Chapter 4 model is very similar, with the important difference that it only uses local factors. This simplify the structure and allow for exact inference. Global dependencies are approximated in our model using using stacking techniques.

In (Niehues and Vogel, 2008), the first group of features are local features which depend only on the source and target words (Figure 2.8(a)). This group includes lexical translation probabilities obtained by IBM model 4; the relative distance of the sentence positions of both words which should help to aligning words that occur several times in the sentence; the relative edit distance between source and target word, which should improve the alignment of cognates; a feature indicating if source and target words are identical which helps aligning dates, numbers and named entities, which are quite difficult to align using only lexical features since they occur quite rarely; finally the predictions of IBM4 are also used as features. See also the discussion about possible features in Section 2.6.

The second group of features are the fertility features. The corresponding factored node for a source word is connected to all M random variables representing the links to the target words, and the node for a target word is connected to all the N nodes for the links to source words (Figure 2.8(b)). Indicator features for the different fertilities up to 3 are used. Additionally, there is a real-valued feature that uses the IBM4 probabilities for the different fertilities. In our MaxEnt we discretize all real-valued feature and binarize the result. By doing this way, multiple model parameters are used instead of only one parameter in the real-valued case, which seems to yield better performance for our model.

The first-order features model the first-order dependencies between the different links.

They are grouped into different directions. The factored node for the direction (t, s) is connected to the variable nodes $y_{i,j}$ and $y_{i+t,j+s}$. For example, the most common direction is $(1, 1)$, which describes the situation that if the words at positions i and j are aligned, also the immediate successor words in both sentences are aligned as shown in Figure 2.8(c). The directions $(1, 1)$, $(2, 1)$, $(1, 2)$, and $(1, -1)$ are used. So this feature is able to explicitly model short jumps in the alignment, like in the directions $(2, 1)$ and $(1, 2)$ as well as crossing links like in the directions $(1, -1)$.

Gradient descent methods are used with two different objectives: the log-likelihood of the data and an approximation of the AER or the F-score (Fraser and Marcu, 2007b) using a sigmoid functions as in (Gao et al., 2006; Suzuki, McDermott, and Isozaki, 2006). The sigmoid approximation is needed since the AER and F-score can not be differentiated which is necessary for gradient-based training. The AER objective enables the training to use from data annotated with sure and possible links, for which the likelihood objective is not sensible. The advantage of the F-score is that there is an additional parameter α, which allows to bias the metric more towards precision or more towards recall. Optimization towards AER is also used in other discriminative approaches such as boosting (Wu and Wang, 2005).

2.3.4.2 Other models

Model 6 introduced by Och and Ney (2003) can be seen as the first approach in which IBM model alignments have been combined in a log-linear fashion.

Cherry and Lin (2003) propose a discriminative model which uses link probabilities as in the weighted matrix but augment it with global context features. Search is then performed using greedy best-first search under one-to-one and cohesion constraints. Liu, Liu, and Lin (2005) incorporate various global features derived from other sources into a globally normalized conditional model:

$$p(\mathbf{A}|\mathbf{e}, \mathbf{f}) = \frac{1}{Z(\boldsymbol{\theta}, \mathbf{e}, \mathbf{f})} \exp \boldsymbol{\theta}^\top \mathbf{g}(\mathbf{A}, \mathbf{e}, \mathbf{f}) \tag{2.23}$$

with a simple decision rule that does not require the normalization factor. Feature functions used here are IBM3 probabilities, PoS tags and bilingual dictionaries. Inference uses a greedy search algorithm based on a heuristic gain function that can be computed incrementally. They use an iterative scaling algorithm for parameter estimation based on an n-best list of highly probable alignments.

Moore (2005); Moore, Yih, and Bode (2006) introduce a similar framework using linear combination of features but drop the probabilistic interpretation and get rid of the normalization constant. Search is not trivial and includes a beam search strategy. To avoid preference to alignment with many links which stems from the simple sum over features, only 5 alignment patterns are allowed (1-1, 1-2, 1-3, 2-1 and 3-1). Additionally, links need to include the strongest individual association for at least one token pair. This corresponds to a greedy selection with respect to association scores. Training is performed using the averaged perceptron Collins (2002). A similar model, with hierarchical search using syntactic parse trees is proposed in (Riesa and Marcu, 2010) which is also trained using the averaged perceptron.

Venkatapathy and Joshi (2007) propose discriminative re-ranking approach which enables to make use of structural features effectively. The alignment algorithm first generates a list of n-best alignments using local features. Then it re-ranks this list using global features. All the n-best alignments are used to update feature weights during parameters estimation through Margin Infused Relaxed Algorithm (MIRA) (Crammer et al., 2006) unlike Moore, Yih, and Bode (2006) where only the best alignment is used.

All these methods needs to enumerate all possible alignments during parameter estimation. However, there is no efficient inference algorithm for global optimization with models that include arbitrary global features. A compromise is done in (Ayan, Dorr, and Monz, 2005;

Ayan and Dorr, 2006a) where each link in the matrix is modeled separately using a *neural network* or MaxEnt with strictly local features and global constraints being ignored.

This section presented several methods to obtain symmetric alignments by using global functions to score alignment matrices. Such models are able to take link dependencies into consideration, but at a high computational cost, especially for long-distance interactions. This is mainly because no restriction on the alignment space is imposed. In the next section, we will discuss how the SCFG formalism can be used to model the equivalence between two sentences using a constrained alignment space. Such constraints reduce the complexity while accounting for long-distance interactions.

2.4 Syntactic and Hierarchical Alignments

We now consider an additional type of constraints used in *Tree alignment*, which is a special case of structure alignment where the output **a** must be a strictly compositional, hierarchical alignment (Wu, 2010). Each sentence has a hierarchical structure represented as a parse tree, where every subtree spans a part of the sentence.

Aligning two subtrees means that words in the yield of the first can be aligned only to words in the yield of the second. This is called the *crossing constraint* (Wu, 2010), and has several benefits. First, the crossing constraint greatly reduces the space of possible alignments and thereby reduces the search complexity; second, due to its relation to syntax, this constraint is accurate most of the times; third, large-distance reordering can easily be modeled while avoiding the complexity of arbitrary permutations. Note that a simple local inversion between constituent in a high level in the hierarchy accounts for a long-distance reordering on the level of leaves.

There exists two alternatives to use such constraints. The first is to separately parse each sentence, possibly with two distinct Context-Free Grammar (CFG)s, and to use a parse-parse-match strategy taking the parse trees as input. However, this approach suffers from the lack of appropriate, robust, monolingual grammars; mismatch of the grammars across languages; and inaccurate selection between multiple possible constituent matchings (Wu, 2010). The second alternative is to simultaneously parse both of the sentences using a synchronous CFG, producing parses for both sides along with the alignment. Obviously, the major disadvantage of such an approach is the difficulty of obtaining the grammar.

Similar to the approaches of Section 2.3, the models we describe here produce many-to-many alignment structures. However, to reduce the number of parameters and the computational complexity they are usually used in one-to-one settings.

2.4.1 Inversion Transduction Grammars

A syntax-directed transduction is a set of bisentences generated by some SCFG (Lewis and Stearns, 1968; Aho and Ullman, 1969). Compared to a Finite-State Transducer (FST), which is the special case with limited expressive power, a general SCFG is more expensive to biparse, train and induce.

The computational complexity for Viterbi chart (bi)parsing, and EM training algorithms for a FST is $O(n^4)$ while it is $O(n^{2n+2})$ for general SCFG. ITG (Wu, 1995a; Wu, 1995b; Wu, 1997) is a special case of SCFG and equivalent to *binary* or *ternary* SCFG whose transduction rules are restricted to straight and inverted permutations only. Such restrictions reduce the computational complexity to $O(n^6)$, and make ITG an attractive intermediate solution whose generative capacity and computational complexity falls in between FSTs and SCFGs. Søgaard (2009) discusses the complexity of the alignment problem within this formalism.

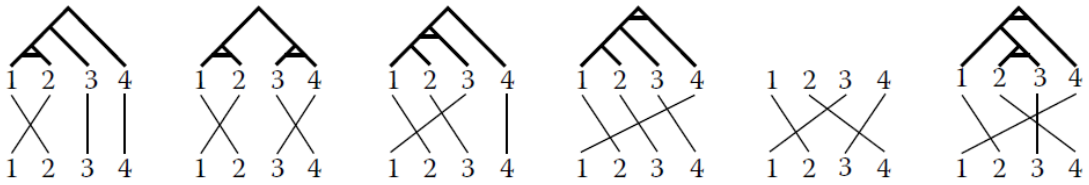


Figure 2.9: *Examples of alignment patterns with ITG parses. One pattern is not attainable which is called* inside-outside *alignment.*

In a 2-normal form ITG, each transduction rule takes one of the following forms:

$$S \rightarrow X$$
$$X \rightarrow [XX]$$
$$X \rightarrow \langle XX \rangle$$
$$X \rightarrow s/\varepsilon$$
$$X \rightarrow \varepsilon/t$$
$$X \rightarrow s/t$$

where [] represents straight rule and <> an inverted rule, s and t are source and target language terminal segments. Non-terminal rules can also be lexicalized (Zhang and Gildea, 2005). ε on both sides accounts for insertion and deletion of tokens.

Although ITGs have proved expressive enough to model most reordering patterns occurring in real data, some patterns are still not attainable. Some of these patterns are shown in Figure 2.9 adapted from (Wu, 1997). Zens and Ney (2003) discuss the expressiveness of ITGs. Beside its expressiveness, the main problem with using ITGs for alignment is that exhaustive biparsing runs in $O(n^6)$ time. Several ways to lower the complexity of ITGs have been suggested. One way is to use pruning methods. For example, Haghighi et al. (2009) do pruning based on the probabilities of links from a simpler alignment model (HMM), which reduces the time complexity by two orders of magnitude. Zhang and Gildea (2005) propose "Tic-tac-toe" pruning, which is based on the IBM1 probabilities of word pairs inside and outside a pair of spans. Zhang et al. (2008) present a method for evaluating spans in the sentence pair to determine whether they should be excluded or not. Their algorithm has a best case runtime complexity of $O(n^3)$. Liu, Li, and Zhou (2010) combine several clues in a discriminative pruning framework. A different approach is taken in (Saers, Nivre, and Wu, 2010). Instead of using full ITGs, they subject the grammar to a linearity constraint where rules may have at most one nonterminal symbol in their production. This constraint reduces the complexity of exhaustive biparsing of a sentence pair to $O(n^4)$. This can be further improved by applying additional pruning. This constraint implies a significant reduction of expressiveness, which does not seem to negatively affect the performance (Saers, Nivre, and Wu, 2010).

2.4.2 parameterization and Learning

In the generative setting, a stochastic context-free grammar associates a probability to every rule in the grammar. The probability of biparse tree is the product of the probabilities of all the rules used in the generation.

Given such a grammar, the task of alignment is cast as a biparsing problem, where the rule probabilities guide the search for the best scoring biparse (Wu, 1995a; Wu, 1995b; Wu, 1997) present a bottom-up parsing algorithm that generalizes the monolingual CYK algorithm to the bilingual case (Stochastic ITGs). Efficient parameter estimation is possible through

inside-outside algorithm (Lari and Young, 1990; Goodman, 1999), which is similar to the forward-backward algorithm for linear chains.

Unsupervised parameter estimation can be performed using EM (Wu, 1995b). Inside-outside probabilities are used to compute expected counts in the E-step, which are then re-normalized in the M-step. Saers and Wu (2009) show that this model produce better alignment than IBM models for German-English, Spanish-English, and French-English Europarl data (Koehn, 2005), in terms of translation quality.

So far, we have discussed a generative parameterization of ITGs, we now move to the discriminative setting. Every alignment is scored with a function that does not necessarily factor in terms of derivation steps (according to a generative story). The ITG is used merely as a constraint on the space of possible alignments. Haghighi et al. (2009) investigate the effect of using ITG constraints in discriminative one-to-one alignments. As already signaled by (Cherry and Lin, 2006b), ITGs have several advantages over the one-to-one constraints in general matching that have been widely used in symmetric discriminative alignments (Melamed, 2000; Taskar, Lacoste-Julien, and Klein, 2005; Moore, Yih, and Bode, 2006). First, the additional structural constraints seem to match the linguist structure. Second, they permit terminals to span several words without increasing the computational complexity, something that general matching can not efficiently do. Third, they admit a range of training options; as with general one-to-one matchings, margin-based objectives can be optimized. However, unlike with general matchings, one can also efficiently compute expectations over the set of ITG derivations, enabling the training of conditional likelihood models.

A major challenge for discriminative training for ITGs is that it requires a corpus annotated with ITG trees. However, manual annotations are often not one-to-one ITG alignments. The recent work of Søgaard and Kuhn (2009); Søgaard and Wu (2009) provides an extensive empirical study on the expressiveness of ITG alignments with respect to their ability to generate manual alignments. Haghighi et al. (2009) illustrate that for gold standards that are outside the ITG class, directly training to maximize the margin is unstable, and training to maximize the likelihood is ill-defined. A solution would be to use *pseudo-gold* alignments with minimal distance from the true reference alignment.

2.4.3 Syntactic Constraints

Most alignment methods use surface statistics as the only information to obtain alignments. However, the success of structural constraints, which are highly related to syntax, motivates the following question: could alignment models benefit from incorporating syntactic and linguistic analysis? After all, syntactic models are increasingly successful in SMT (Yamada and Knight, 2001; Chiang, 2005; Galley et al., 2006), and syntax-directed alignment may be more coherent with such model than general alignments.

Lopez and Resnik (2005) suggest to parameterize the distortion model in the HMM alignment using the "tree distance" between each pair of target words, conditionally on the PoS tag of the previous word: $p(a_j|a_{j-1}) = p(a_j|\tau(e_{a_j}, e_{a_{j-1}}), \text{PoS}(e_{a_{j-1}}))$. Given a dependency parse of the target sentence, the distance τ between two words is defined as the number of links separating them from their closest common ancestor node in the parse tree. This is a way to incorporate PoS features into a generative model, however their obtained results are not superior to the surface statistic and obtaining dependency parses is expensive. Similarly, DeNero and Klein (2007) integrate a target language syntactic parse trees into the transition model of an HMM. The transition probabilities now condition upon paths through the target parse tree, allowing the model to prefer distortions which respect the tree structure. Taking the target language constituent explicitly into account is helpful when used in conjunction with syntax-based translation systems.

An alternative is to use syntactic analysis as hard constraints on possible alignments similar to the ITG-based methods. Crego and Habash (2008) use the result of *chunk* analysis

to post-process the alignments.

A final, arguably simpler, solution is to incorporate syntax-based information into the model in guise of features, which is done in almost every model that uses feature functions to represent data.

2.4.4 Other Syntax-Based Models

Yamada and Knight (2001) presented a tree-to-string alignment model. The model is trained using English syntactic trees generated from a high quality syntactic parser and Japanese strings. A particular generative story applies operations to the English tree to generate the Japanese string, and this induces an alignment. Gildea (2003) extended this model to tree-to-tree alignments.

2.5 Phrase-Based Alignment Models

So far, we have considered alignment with word constraints, where alignable units are words. Unfortunately, single words are not always the best units to capture translation relations. Problems such as word fertility, lexical ambiguity and word order can be solved to a large extent by relaxing the single word constraint and allowing phrases to be aligned instead of words.

Let us recall the definition of a phrase alignment between two sentences to be the set of links between phrases:

$$\mathbf{A} = \{(\mathbf{p}, \mathbf{r}) : \mathbf{p} \subseteq \mathcal{N} \text{ and } \mathbf{r} \subseteq \mathcal{M}\}, \tag{2.24}$$

where $\mathcal{N}$ and $\mathcal{M}$ are the set of source and target indices respectively. Some decisions concerning word lexical ambiguity, fertility and reordering, that had to be made explicitly in word alignment, are partially taken implicitly by considering longer units. Phrase-based models are typically simpler than word-based models, at the cost of increased learning and inference complexity.

2.5.1 Bisegmentation

A bisegmentation is obtained under *bijectivity* constraints for phrase alignments, where each alignment $\mathbf{A}$ implies a phrasal partition of the source and target sentences, along with a bijective mapping between them. The number of such joint phrase segmentations and alignments is exponential in the sentence length, which makes enumerating all of them infeasible. For models that operate on the full phrase alignment space (Marcu and Wong, 2002; DeNero et al., 2006), the computational complexity of inference is NP-hard (DeNero and Klein, 2008). Inference algorithms must either be approximate as in (Marcu and Wong, 2002; Birch et al., 2006), which rely on word alignments to obtain a good starting point for a hill-climbing heuristic in a restricted search space; or require running time exponential in the sentence length as the DP (DeNero et al., 2006) and the Integer Linear Programming (ILP) solutions proposed in (DeNero and Klein, 2008). However, the application of additional restrictions on this combinatorial space can lead to polynomial-time DP solutions. Such restrictions may be *linear* as in monotone and distortion-limited alignments (Zens and Ney, 2004), or *hierarchical* as in ITG alignments (Cherry and Lin, 2007). As discussed in Section 2.4, ITG can straightforwardly include phrase productions in addition to words and still permits polynomial-time exploration of the search space with a complexity $O(n^6)$. However, for large corpora and with long sentences, inference remains prohibitively costly.

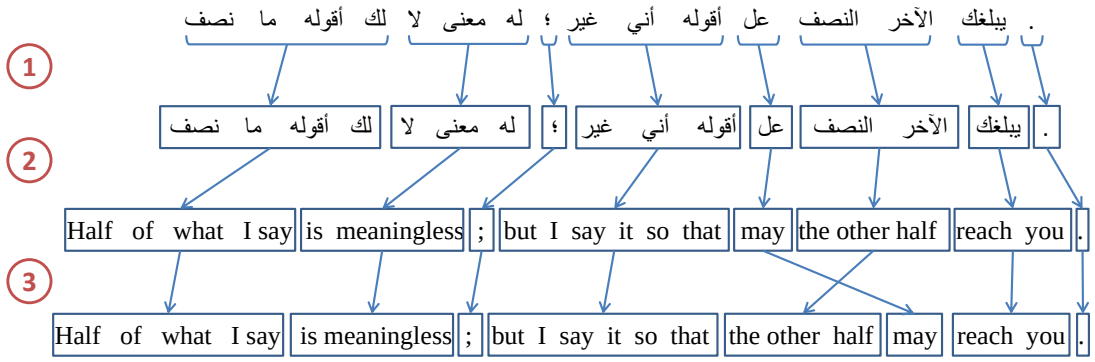


Figure 2.10: *The phrase-based translation model: starts with a segmentation step (1), followed by a lexical substitution step (2), and ends with a permutation step (3).*

2.5.1.1 Generative models

In the generative framework presented in Section 2.2.2, Marcu and Wong (2002); Birch et al. (2006) propose a three step generative process to model the *joint* distribution $p(\mathbf{A}, \mathbf{e}, \mathbf{f})$. First, the number ($n$) of phrase pairs is chosen; then n phrase pairs are drawn independently from a distribution over phrase pairs; and finally, phrase pairs are reordered. Therefore, an alignment implies a joint segmentation of the source and the target sentences, and a permutation of the resulting phrases on one side. DeNero et al. (2006) propose a similar generative process to model the *conditional* distribution $p(\mathbf{A}, \mathbf{e}|\mathbf{f})$. This is illustrated in Figure 2.10.

Similar to word-based models, parameter estimation can be performed by MLE using EM. However, computing expectations requires to sum over all bijective phrase alignments, which is intractable. Therefore, (Birch et al., 2006; DeNero et al., 2006) and similar approaches (Cherry and Lin, 2007) use ITG constraints in addition to word alignment based pruning. ITG are interesting also for syntactic alignment and have been used to align spans in a source sentence to nodes in a target parse tree (Pauls et al., 2010).

Hidden semi-Markov models An alternative approach to the conditional generative model for phrase alignment is based on an extension of standard HMMs, presented by Ostendorf, Digalakis, and Kimball (1995).

Deng and Byrne (2005) describe a word-to-phrase HMM which modifies the parameterization of the traditional word-based HMM model to allow a state to produce more than one words. Figure 2.11, adapted from (Deng and Byrne, 2008), shows such an alignment for a Chinese-English sentence pair. However, this model only changes the parameterization and not the set of possible alignments. This model provides a more powerful formulation of a phrase length model than the "stay" (loop) probabilities in word-based HMM alignment (Toutanova, Ilhan, and Manning, 2002). Andrés-Ferrer and Juan (2009) use a similar model interpolated with IBM1 when only monotonic alignments are allowed.

The degeneracy problem Unfortunately, MLE training for phrase models often lead to degenerate solutions for both the joint and the conditional generative models (DeNero et al., 2006):

- The likelihood can be artefactually increased by using fewer multiplicative terms, which can be achieved by selecting large phrases in order to explain the training data. As

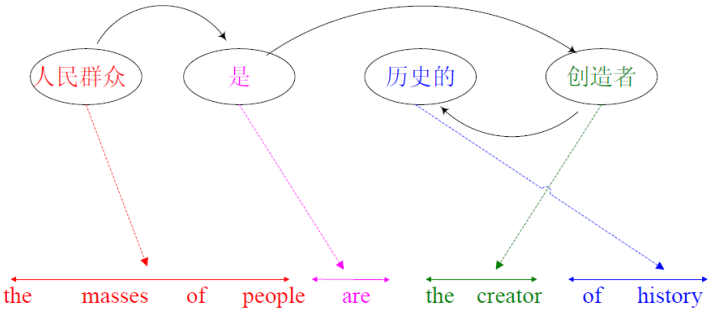


Figure 2.11: *An example of word-to-phrase HMM Alignment (Deng and Byrne, 2008). Source words are treated as states and target phrases as observations.*

a result, the joint model often fails to learn to translate individual words and short phrases.

- Imposing competition between segmentations may lead to spurious solutions to the translation lexical ambiguity under the conditional model. For instance, the French "une note" can be translated into English as "a note" or as "a grade". Using these two parallel sentences MLE could choose the parameters $p(\texttt{note}|\texttt{a note}) = 1$ and $p(\texttt{grade}|\texttt{note}) = 1$ which maximize the likelihood by conditioning on rare phrases in low-entropy distributions.

Several solutions to the degeneracy problem have been investigated. Moore and Quirk (2007) proposed a new conditional model that does not cause large and small phrases to compete for the same probability mass. May and Knight (2007) added additional model terms to balance the cost of long and short derivations in a syntactic alignment model. Bansal, Quirk, and Moore (2011) combine the phrase-based HMM model of (Andrés-Ferrer and Juan, 2009), without the monotonicity requirement, and agreement constraints of (Liang, Taskar, and Klein, 2006) (cf. Section 2.3.1.2). Phrases may be used in both the state and observation space of both sentences, hence agreement during EM training no longer penalizes phrasal links. Agreement constraints help avoiding the degeneracy problem since meaningful phrasal links that are likely in both alignment directions will be reinforced, while phrasal links likely in only one direction will be disregarded.

2.5.1.2 Bayesian models

Many of the previous solutions to the degeneracy problem integrate the prior knowledge as constraints on the search space. Alternatively, *Bayesian priors* incorporate such knowledge into the model. Bayesian modeling treats model parameters as additional random variables that have associated distributions. This additional distribution over parameters adjusts the learning objective while maintaining the same structure and parameterization for the underlying model.

Introducing Bayesian priors to the generative models encodes a preference for short phrases rather than long ones; and a preference for reusing phrases across the entire corpus. To express these priors, DeNero, Bouchard-Côté, and Klein (2008) use a *Dirichlet Process*, which is a prior over multinomials with an unbounded number of dimensions, and *collapsed Gibbs sampling* which is an approximate inference technique with desirable convergence properties. Instead of using word alignment as constraints, they can be used for initialization (DeNero, Bouchard-Côté, and Klein, 2008). In a similar view, Zhang et al. (2008) reduce the complexity by using IBM1 scores in dynamic pruning algorithm (Zhang and Gildea, 2005). They also incorporate a sparse prior using Variational Bayes EM to avoid overfitting.

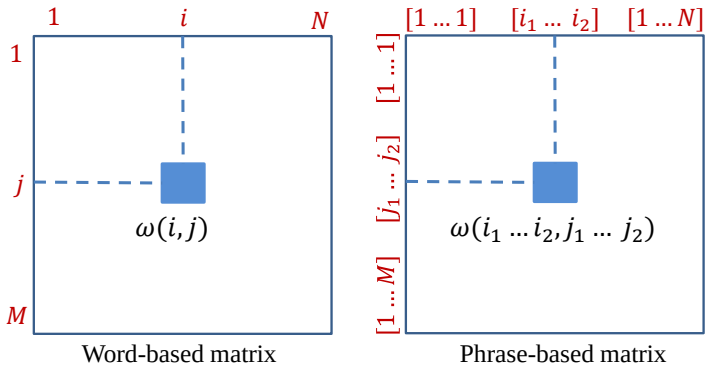


Figure 2.12: *Comparison between word-based and phrase-based matrices.*

Blunsom et al. (2009) use a non-parameteric Bayesian formulation to include a hierarchical prior. They use a Gibbs sampler for approximate inference over the infinite space of possible translation units. Unlike many other previous approaches, they do not use heuristics pruning or constraints from word alignments.

2.5.1.3 Discriminative models

Alternatively, these models can be trained discriminatevly in a supervised way, as in Haghighi et al. (2009), who describe a block ITG model in addition to the word-based model.

Liu, Li, and Zhou (2010) propose a discriminative pruning framework for discriminative ITG. The pruning model uses a log-linear model to integrate several features (like Model 1 probability and HMM posteriors) that help identify the correct span pair and is trained using Minimum Error-Rate Training (MERT) (Och, 2003). On top of the discriminative pruning method, a discriminative ITG that incorporate hierarchical phrases is trained. Features computed on such phrases are combined in a log linear model similar to (Liu, Liu, and Lin, 2005; Moore, 2005).

2.5.2 Generalized Phrase Alignment

In previous sections, we have described word-based and bijective phrase-based alignment. We now describe methods that relax the bijectivity constraints and results in *overlapping* phrases that do not necessarily form a partition. Hence, the focus is on the extraction of reliable translation equivalents, sometimes called *translation spotting* (Véronis, 2000).

2.5.2.1 Extraction heuristics

In a similar way to the weighted word-based matrix, we can represent the space of all possible phrase alignment using a binary matrix of dimensions $2^N \times 2^M$, where rows and columns are indexed with *sets* of source and target positions. We restrict the matrix to contiguous phrase pairs only ($\mathbf{p} = i_1 ... i_2, \mathbf{r} = j_1 ... j_2$). The matrix alignment function maps all possible phrasal links to the binary active/inactive set: $\forall \mathbf{p}, \mathbf{r} : \quad A_{\mathbf{p},\mathbf{r}} \in \{0, 1\}$.

Several phrase alignment methods start by associating a score $c_{\mathbf{p},\mathbf{r}}$ to each cell in the matrix. Figure 2.12 depicts such a matrix. Any number of alternatives can be used to score the phrasal links.

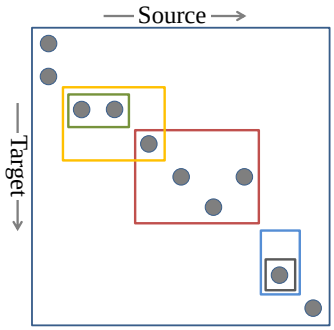


Figure 2.13: *Phrase pairs consistent with the word alignment.*

The standard approach Koehn, Och, and Marcu (2003) compute first a word or a phrase alignment and use it to induce a binary score $c_{\mathbf{p},\mathbf{r}} \in \{0, 1\}$:

$$c_{\mathbf{p},\mathbf{r}} = \begin{cases} 1 & \text{if } (\mathbf{p}, \mathbf{r}) \text{ is consistent with the word alignment} \\ 0 & \text{otherwise} \end{cases} \tag{2.25}$$

In order for a phrase pair to be consistent, it should contain at least one word-based link; and no word inside it is aligned to a word outside it. Figure 2.13 shows few examples of consistent phrase pairs.

The major issue of this heuristic is its sensitivity to word alignment errors. Since the consistency constraint is based on the "Viterbi" alignment, an error could prevents the extraction of many correct phrase pairs.

Weighted phrase-based matrix To alleviate the problem of the standard extraction approach, the strict consistency constraint can be replaced by a more informative one which may go beyond the "Viterbi" alignment. For instance, one can compute smoothed scores $c_{\mathbf{p},\mathbf{r}} \in [0, 1]$ to evaluate each phrase pair. Some filtering techniques multiply the binary consistency score by a probability resulting from statistical significance tests (Johnson et al., 2007; Tomeh, Cancedda, and Dymetman, 2009). Vogel (2005) use a linear combination of features computed from a weighted word-based matrix populated with IBM1 scores. Similarly, Liu et al. (2009) use the product of two scores that characterize the consistency based on a weighted word matrix built from a set of N-best alignments. Zettlemoyer and Moore (2007) use the same combination of features used by a phrase-based translation system (Koehn, Och, and Marcu, 2003), tuned with MERT (Och, 2003). Deng and Byrne (2005) combine two phrase alignment posteriors, computed under two HMMs, one for each directions. Venugopal, Vogel, and Waibel (2003) apply several features but use a weighted linear combination. One issue with these method is that no learning is involved to weight the combined features or to select the threshold. Therefore, Deng, Xu, and Gao (2008) propose to tune the weights of this model by plugging it into an end-to-end translation pipeline and by maximizing BLEU.

Once the scores are in place, simple thresholding similar to MBR decoding can be applied to obtain the final alignment (Koehn, Och, and Marcu, 2003; Venugopal, Vogel, and Waibel, 2003; Johnson et al., 2007; Tomeh, Cancedda, and Dymetman, 2009). The presence of this threshold allows the extraction procedure to control the balance between precision and recall. Vogel (2005) extracts maximum scoring phrase pairs for source phrases. Zettlemoyer and Moore (2007) use a competitive linking algorithm similar to (Melamed, 2000).

2.5.2.2 Translation spotting

Various simple, techniques can be applied for lexicon extraction and translation spotting. (Tiedemann, 1999; Tiedemann, 2003a) use smaller aligned segments to iteratively reduce the

size of unaligned longer segments. For example, many sentence-aligned bitexts include very short sentence fragments, and their alignments can often be used immediately as lexical translation equivalent. These initial entries can then be used to mark other occurrences of known equivalence pairs in the bitext. Lardilleux and Lepage (2008) define an alignment method that relies on simple heuristics based on similarities and differences between sentences.

2.5.2.3 Discriminative models

Deng, Xu, and Gao (2008) represent the extraction of phrase pairs as a binary classification problem where each classification decision is made independently. A linear model is used to combine several features, of which the weights are learned to maximize BLEU. The threshold used to select phrase pairs is considered as a parameter and is optimized with the feature weights. The major issue with this approach is its complexity: given one set of parameters, a phrase table is built and used to compute the BLEU score on some corpus. This requires constructing and training a translation system including tuning the weights of its features. This is needed many times during training which becomes prohibitively expensive. A sub-optimal compromise is to discard the tuning of the translation system's weights and fix them once and for all.

All the previous approaches to general phrase alignments consider each phrase pair independently from the others. Therefore, DeNero and Klein (2010) recast the problem as a structured classification problem, in which a complex object containing all extracted phrase pairs (called the extraction set) is predicted for an input sentence pair. They use a discriminative linear model to score the set of extracted phrase pairs. Similar to previous approaches, features on phrase pairs can be easily incorporated. The used loss function is a phrase-level F-measure which requires hand-annotation of extraction sets. This is problematic since only word annotations are typically available. To solve this issue, a deterministic mapping form the word alignment to the extraction set is defined and used to obtain training annotations. Inference in the extraction set space is intractable: the model does not factor over disjoint word-to-word links or minimal phrase pairs, and so existing inference procedures do not directly apply. A solution is to use ITG constraints and resort to a DP algorithm, originally presented in (Haghighi et al., 2009), and which can be augmented to score extraction sets that are indexed by underlying ITG word alignments.

The main advantage of this method is the modeling the interactions between extracted, overlapping phrase pairs. However, the loss function is not directly related to the translation quality.

2.6 Features

When making alignment decisions between two sentences, the word sequences themselves $\mathbf{e}$ and $\mathbf{f}$ and any number of external information regarding the context may be relevant. The context may include resources external to the sentences such as the output of taggers and parsers or any other type of annotation including the output of other alignment methods. It is helpful to transform such input data into a reduced representation set of features. The space of input-output pairs $\Sigma^* \times \Lambda^* \times \mathcal{A}$ is mapped to a d-dimensional $\mathbb{R}$ space through a feature vector function, $\mathbf{g}(\mathbf{A}, \mathbf{e}, \mathbf{f})$. Each feature function g_k maps a sentence pair and its alignment to a real value. If features engineering is carefully done, features set will extract all the relevant information to perform the alignment, hence it can be seen as some kind of dimensionality reduction.

Designing a feature is guided by what kind of information it captures, its scope and its representation. We discuss these aspects in this section.

2.6.1 Type

Typical feature functions in Natural Language Processing (NLP) perform some symbolic test on the given context and return a binary value indicating the success status of the test. For instance, a feature based on word lexical content may take the form:

$$g_{\text{lex}}(i, j, \mathbf{e}, \mathbf{f}) = \begin{cases} 1 & \text{if } f_i = \texttt{le} \wedge e_j = \texttt{the} \\ 0 & \text{otherwise} \end{cases} \tag{2.26}$$

Features functions can also return discrete values such as the distance from the diagonal of the alignment matrix: $g_{\text{lex}}(i, j, \mathbf{e}, \mathbf{f}) = |i - j|$. It is however a common practice to *binarize* such features by incorporating the return value into the test itself. So a the previous feature is re-expressed using a separate feature per distance $0, 1, 2, ...$:

$$g_{\text{lex}}(i, j, \mathbf{e}, \mathbf{f}) = \begin{cases} 1 & \text{if } |i - j| = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.27}$$

Similarly, general real-valued features can be *discretized* and then binarized, which practically results in superior performance in many cases.

2.6.2 Indicators of alignment

Various types of information are good indicators for alignment and can be explored by the features (Wu, 2010; Tiedemann, 2011). Here are few examples.

Lexical information. Succeeding in finding some lexical link between source and target segments may serve of strong indication of a translational equivalence relation. Suffixes and prefixes of the linked words may indicate some derivational similarities. Bilingual dictionaries and wordnets, which are large lexical databases of nouns, verbs, adjectives and adverbs, grouped into sets of cognitive synonyms, can be used for lexical matching. Matching between cognates, which are words that have a common origin across languages, is also helpful for related languages. Possible lexical matches can be found by measuring string distance in languages with similar alphabets such as French and English. Examples include named entity ("Saddam Hussein" - "Sadam Hussayn") and numerical items ("3,14" - "3.14"). Word that have a common etymological origin can also be matched. An example in Indo-European languages, the words "night" (English), "nuit" (French), "Nacht" (German) "nacht" (Dutch) are cognates. The Hebrew "shalom", the Arabic "سلام salam", the Maltese "sliem" and the Amharic "selam" (peace) are also cognates, derived from a Proto-Semitic root. However, words or phrases that look or sound similar but differ in meaning (false friends) may be misleading. An example is Portuguese "raro (rare)" vs. Spanish "raro" (strange).

Segment length. Word or phrases that express the same meaning tend to have similar lengths, as measured by the number of words of characters. Such correlation is notably high in similar languages and much less reliable for distant languages.

Position. Alignments are mainly monotonic, centered around the diagonal of the alignment matrix.

Distributional profiles. Corresponding words and segments usually have similar distributional properties across the corpus, which can be measured by means of statistical tools.

This is especially helpful in case of rare words and hapaxes that would be otherwise difficult to align.

Linguistic features. Annotating the parallel corpus with morphological and syntactic information, possibly with the help of external tools, make the alignment benefit from similarities beyond surface statistics. For example, aligning two verbs is more probable than aligning a verb to a noun.

Such indicators define the type of information that we would like to incorporate into the feature. From which context these information are extracted is defined by the scope of the feature which we discuss next.

2.6.3 Scope

We can divide features into three types according to the portion of the alignment structure being considered (Tiedemann, 2011):

- **Local features:** They restrict their context to the current link $\mathbf{g}(i, j, \mathbf{e}, \mathbf{f})$. Instances of such features are abound in the literature since they are the less expensive. Examples include lexical content, prefix of suffix of the connected words; association scores; co-occurrence information; position in the alignment matrix; string similarity; and any number of the cues we discussed in the previous section.
- **Dynamic features:** If the alignment structure is predicted sequentially, a *history-based* approach can be used to keep track of previously predicted links in addition to the current one. An example of such feature is described by Ittycheriah and Roukos (2005).
- **Global features:** The entire alignment structure is taken into consideration at once. For instance the number of links included in the alignment, or the score given to the alignment by another alignment model. Such features can be found in (Liu, Liu, and Lin, 2005; Moore, 2005) and many more.

When designing a feature, independence assumptions about modeled variables are needed to maintain tractability. With larger contexts (from local to global), comes sparsity issues which affects the parameter estimation. Moreover, computing a feature of the modeled variable, is exponential in time and space in the size of the context. However, features of observed variables are less expensive and can be computed linearly in the size of the context. This is the case of the feature:

$$g_{\text{lex}}(i, j, \mathbf{e}, \mathbf{f}) = \begin{cases} 1 & \text{if suffix}(f_{i-1}) = \texttt{ing} \wedge \text{POS}(e_{j-1}) = \texttt{VERB} \\ 0 & \text{otherwise} \end{cases} \tag{2.28}$$

2.7 Summary

In this chapter we have presented a survey of approaches to the problems of word and phrase alignment.

In word-based models, we have first considered the IBM models which date back to the early days of word-based translation systems. Despite their numerous shortcomings, these models are still widely used in practice, especially as a first step in training phrase-based translation systems. This is mainly because they are trained in an unsupervised manner from a sentence-aligned parallel corpus. However, they have several drawbacks. Incorporating additional features is not straightforward in generative models; large amount of training data is required in order to obtain reasonable results; the likelihood objective does not relate directly to the quality of the alignment; and finally, they can only produce one-to-many, asymmetric alignments. IBM models belong to a family of approaches that considers the alignment as a sequence labeling problem. Within the same family, we have described an approach,

which addresses the issue of incorporating features into generative models. This is done using an alternative local log-linear parameterization. Then, we have described discriminative approaches to one-to-many alignments, such maximum entropy models. Discriminative models are are trained from supervised data. The are typically simpler than their generative counterparts, since they only model the alignment variable. Additionally, they achieve a competitive performance using relatively smaller amounts of training data.

After describing one-to-many alignments, we have moved to another family of techniques, which does not suffer from asymmetry. We have first described a heuristic which obtains symmetry by combining two directional alignments. It makes symmetrization decisions so as to balance the precision and the recall of the resulting alignments. This heuristic performs surprisingly well in practice and is used by current state of the art translation systems. Nevertheless, model-based approaches to the alignment combination problem typically outperform the heuristic. We have therefore discussed approaches that perform the symmetrization during the training of the directional models instead of during the inference. This is done using agreement constraints. We have also discussed how multiple features can be incorporated to the alignment combination process by using discriminative models to make the combination decisions. After having discussed the combination methods, we have presented techniques that directly produce symmetric alignments. One such approaches is based on weighted alignments matrices. First, an association score is computed for each possible link in the alignment matrix; then, an alignment is obtained either by thresholding the scores, or by performing a search for the best scoring alignment under some constraints. The thresholding method is quite popular because it is simple, and it enables to control the balance between precision and recall. Whereas many search methods are computationally costly and typically rely on approximations. From matrix-based approaches we have moved to more general methods, which used a discriminative function to score the alignments. The advantage of these methods is that global features can be used to incorporate information about the interaction between the links within the alignment. However, a compromise between expressiveness of the model and its computational complexity has to take place, in order to maintain the tractability of the search.

Intractability is the major issue in global approaches. The complexity arises from allowing arbitrary alignment in the goal of capturing long-distance interactions between links. We have therefore described a middle-ground, tractable solutions, based on ITG constraints. Restricting the alignments to have an ITG structure permits long-distance interactions while drastically reducing the number of permissible alignments, and hence reducing the complexity. Although a lot of work has been done around ITG alignments, their use is still limited in practice for two reasons. First, even though ITGs admit polynomial time training and inference ($O(n^6)$ and $O(n^4)$ for linear ITG), they are still prohibitively costly in practice especially for long sentences. Pruning is therefore always required. Second, ITGs do not cover all patterns found in manual alignments which means that some correct alignment can not be obtained. This is problematic especially for discriminative models, trained from gold standards which may not belong to the ITG class.

The main weakness of all word-based models is their incapability to model multi-word phrase alignment explicitly. Many of word alignment difficulties, such as lexical ambiguity, word fertility and word reordering, can be implicitly accounted for in a phrase-based model. After having described word models, we have presented several phrase-based models, in which phrases can be aligned directly as a whole. The first family of phrase alignment models seek to produce a bisegmentation of the parallel sentence. Generative models are frequently used for this purpose. However, these models often get trapped in degenerate solutions. As a remedy, the Bayesian framework can be used in order to incorporate a prior knowledge and guide learning to desired solutions. The second family of phrase alignments is more general. Instead of seeking a bisegmentation, generalized phrase alignments seek to identify all possible phrase correspondences within a sentence pair. Generalized phrase alignments

are directly used to train phrase-based translation systems, and therefore they are of great importance. In practice, the predominant method to obtain generalized phrase alignment is a heuristic, which extracts all phrase pairs that are consistent with an underlying word alignment. A generalization of this heuristic uses a softer definition of the consistency. This enables a better control of the balance between the precision of extracted phrase pairs and their recall. Finally, we have described a discriminative model that scores entire sets of extractable phrase pairs for each sentence pair, and search for the best scoring one. This approach is more directed toward the translation as the final application the extraction sets; and it also takes into consideration the interactions between phrase pairs. However, the computational complexity involved in scoring such sets and searching among them is prohibitive to its use in practice. Moreover, training the model requires gold standards which can not be easily obtained for extraction sets.

Finally, we have discussed several aspects of the feature functions, which are used by the alignment models that we have presented. These aspects have covered the information that helps making the alignment decisions, and how it can be represented in the model. We have also discussed the scope of these features and its impact of the complexity of the model.

CHAPTER 3

Phrase Based Statistical Machine Translation

Machine Translation (MT) is the sub-task of NLP addressing translation from one natural language to another using machines. Over recent years, the field of machine translation witnessed tremendous changes. Nevertheless, the long-standing debate on the feasibility of the ultimate goal of "fully automatic, high quality machine translation" (Bar-Hillel, 1964) continues with a better understanding of the limits of automatic translation (Madsen, 2009).

Similar to many other NLP applications, due to the coupling of powerful machine learning methods with the increasing availability of computational power and necessary resources, translation has been rapidly dominated by statistical approaches, with an unprecedented practical success. Such success can be attributed to several factors. On the one hand, Internet facilitates the *dissemination* and *assimilation* of information from multilingual sources of information: several governments and agencies broadcast multilingual documents, which are accessible to SMT practitioners; moreover, online translation services are nowadays widely used in everyday communication. On the other hand, rapid development in hardware and computing technologies makes it possible to benefit from the growing body of available texts. Additionally, the development of automatic translation metrics and of several free and open source SMT toolkits, facilitate the implementation and the evaluation of translation systems.

Translation is a complex process involving a large number of interacting factors. A *translation equivalence model* attempts to disentangle these various factors, to describe them individually, and to model their interactions. According to such model, translating a text amounts to segmenting it into smaller text fragments (translation units), translating them atomically and recombining their translations afterward. Statistical approaches aim to learn such segmentation, translation and recombination decisions by observing them in large collections of previously translated texts. In most cases, these decisions are implicit in the bitext and can not be observed directly. At this point, the task of bitext alignment is of a great importance to reveal the hidden relations and state them explicitly.

While SMT systems share the same foundations, they diverge in several aspects. The first aspect is the *translational equivalence model* which specifies the formal process for translation decision making. Most systems rely on concepts from automata and formal language theory to perform this modeling. The second is the *parameterization* of this model which is required to score competing translation alternatives and to resolve ambiguities. The parameterization defines a set of statistics (parameters) that are learned from data using machine learning techniques through *parameter estimation*. Third, *decoding* aims to search for the best scoring

translation of a given source sentence according to the model.

In this chapter we give a brief introduction to the *phrase-based* paradigm to Statistical Machine Translation (PBSMT), in which we have performed our experiments presented in this dissertation. For more details on phrase-based SMT and for overviews of other approaches one can refer to several surveys or books covering SMT (Knight and Marcu, 2005; Lopez, 2008a; Koehn, 2010); and related fundamental research in NLP (Manning and Schütze, 1999; Jurafsky and Martin, 2008), artificial intelligence (Russell and Norvig, 2009), and machine learning for NLP (Smith, 2011), and formal language theory (Hopcroft, Motwani, and Ullman, 2006).

3.1 Phrase-Based Translation Model

Phrase-based models translate several contiguous word tokens as an atomic unit, called a *phrase*[1]. *Phrases pairs* that are translation of one another constitute the model's bilexicon and they are stored in a structure famously referred to as the *phrase table*.

The first SMT systems were word-based (Brown et al., 1993) meaning that they used words as the units of translation. However, shifting from words to phrases is advantageous in several ways. We consider the following example. The Arabic collocation "جدير بالذكر" usually translates to English as "worthy of mentioning". The word-based model should choose a fertility of one for "جدير" and translate it as "worthy", a fertility of two for "بالذكر" and translate it as "of mentioning", and then invert their translations. This process involve many decisions that can be avoided in a phrase-based model which can perform the translation directly in one step. Since phrases can have variable length, null translation and fertility are no longer required and many local reordering decisions are made implicitly. Incorporating phrases results in simpler models, yielding fewer decisions to make and hence fewer chances for committing errors. Larger local context also helps dealing with lexical ambiguity. In the case of phrasal verbs such as the Arabic verb "أغار", identifying the meaning requires consulting the following proposition: while "أغار إلى" translates to English as "to help", "أغار على" translates as "to kill". Translating idiomatic expressions and non-compositional phrases becomes feasible by memorizing their translations, as with the Arabic expression "عاد بخفي حنين", which literally translates to "he returned with Hunain's shoes" while it should be translated to "he returned empty-handed".

According to the phrase-based model (Zens, Och, and Ney, 2002; Koehn, Och, and Marcu, 2003; Och and Ney, 2004), translation is performed in three steps that can be implemented by a cascade of finite state transducers (Kumar, Deng, and Byrne, 2006): a **segmentation** step, where the source sentence is first split into disjoint contiguous phrases; a **lexical translation** step, in which each source phrase is translated; and finally a **reordering** step, in which target phrases are permuted into their final order.

Phrase-based translation is implemented in the open source toolkit Moses (Koehn et al., 2007)[2]. Many variants of the phrase-based model have been investigated in the literature.

[1] In this context, the term "phrase" has no specific linguistic meaning.
[2] The Moses toolkit is available at http://www.statmt.org/moses.

Och and Ney (2004) present an alignment template approach that model word reordering based on their part-of-speech categories. Mariño et al. (2006) refer to phrase pairs as tuples and estimate the translation model as *n-gram* distributions over tuples. Other phrase-based variants (Simard et al., 2005; Crego and Yvon, 2009; Galley and Manning, 2010) offer the possibility for phrases to contains gaps that are filled with other phrases during decoding.

While phrase-based models produce better results than their word-based counterparts, they still have issues with the modeling of reordering. Accounting for long-distance reordering is complicated, and distinguishing correct reordering patterns is a challenging task, which may benefit from incorporating syntax constraints. *Hierarchical* and *synchronous context-free grammar* models handle this problem in a more principled way, using more expressive models, belonging to the class of context-free grammar (CFG). These models are closely tied to a linguistic representation of syntax and can better model long-distance reorderings.

3.2 Modeling and Parameter Estimation

Translational equivalence models make it possible to enumerate all structural relationships between pairs of strings. However, the ambiguity of natural language results in a very large number of possible target sentences for any input source sentence. These hypotheses need to be ranked; for this purpose it is customary to assign a real-valued score to any pair of source and target sentences. As in typical statistical decisions problems, we are given an input sentence $\mathbf{f}$, and the goal is to find the best translation $\mathbf{e}$.

Therefore, a function $\omega : \Sigma^* \times \Lambda^* \rightarrow \mathbb{R}$ that maps input and output pairs in a real-valued score, is used to rank possible outputs. Given an appropriate parameterization, this scoring function can be interpreted as the conditional probability $p(\mathbf{e}|\mathbf{f})$ where $\mathbf{e} = (e_1, \ldots, e_M)$ and $\mathbf{f} = (f_1, \ldots, f_N)$ are represented with random variables.

In the FST model of translation, each sentence $\mathbf{e}$ can be derived from $\mathbf{f}$ in several ways according the alignment $\mathbf{d}$ established between source and target words or segments. The value of $p(\mathbf{e}|\mathbf{f})$ is therefore obtained by summing the probabilities of all derivations $\mathbf{d} \in \mathcal{D}$ that yield $\mathbf{e}$.

$$p(\mathbf{e}|\mathbf{f}) = \sum_{\mathbf{d} \in \mathcal{D}} p(\mathbf{e}, \mathbf{d}|\mathbf{f}). \tag{3.1}$$

However, this sum involves an exponential number of terms and hence, a common practice is to resort to directly maximizing the function $p(\mathbf{e}, \mathbf{d}|\mathbf{f})$. The parameters of $p(\mathbf{e}, \mathbf{d}|\mathbf{f})$ are estimated from a parallel corpus using machine learning techniques.

As a side note, one should realize that any of the alignment model, surveyed in Chapter 2, that scores both of the alignment and the target sentence, can be used for translation.

3.2.1 Discriminative Translation Models

Discriminative models are more suitable for translation prediction because they do not try to model the source sentence which is always considered given. In SMT, a popular approach is to use a linear model (Berger, Pietra, and Pietra, 1996; Och and Ney, 2002), as in Equation (3.2):

$$p(\mathbf{e}, \mathbf{d}|\mathbf{f}) = Z(\mathbf{f}, \lambda)^{-1} \exp \sum_{k=1}^{K} \lambda_k h_k(\mathbf{e}, \mathbf{d}, \mathbf{f}), \tag{3.2}$$

where $\{\lambda\}_1^K$ are the scaling factors, associated to the feature functions $\{h\}_1^K$, and $Z(\mathbf{f}, \lambda) = \sum_{\mathbf{e},\mathbf{d}} \exp \sum_{k=1}^{K} \lambda_k h_k(\mathbf{e}, \mathbf{d}, \mathbf{f})$ is a normalization factor required only to make the scoring function a well-formed probability distribution. Fortunately, we can ignore this normalizer

during decoding because it is constant for any given $\mathbf{f}$. Its computation may or may not be required during parameter estimation, depending on the algorithm.

3.2.2 Bilexicon Induction

The hypotheses translations for a given input sentence are constructed from precomputed set of phrase pairs, called the *bilexicon*. The bilexicon is built from a sentence-aligned parallel corpus in one of two ways. Typically, a general phrase alignment (an extraction set) is computed for each sentence pair (cf. Section 2.5.2), and the extracted phrase pairs are accumulated over the entire corpus. This method performs very well in practice and is used in most state-of-the-art translation systems. Alternatively, the bilexicon can be built by harvesting the parameters of a generative translation model that includes a hidden phrase alignment variable. (DeNero, Bouchard-Côté, and Klein, 2008; Saers and Wu, 2011). This approach is less common in practice mainly because training a generative phrase based model is difficult (cf. Section 2.5.1.1).

For each phrase pair in the bilexicon, a set of feature functions are computed and used to score translation hypotheses. We discuss the most commonly used feature functions in Section 3.2.3, and the data structure used to store them, called the *phrase table*, in Section 3.2.4.

3.2.3 Features

A feature can be any function from $\{\Sigma^*, \mathcal{D}, \Lambda^*\} \rightarrow [0, \inf)$ that maps a pair of source and target sentences to a non-negative value. Each feature function typically decomposes in terms of local evaluations at the level of words and also phrases. We now briefly describe the "standard" features introduced in Koehn et al. (2007) and found in other approaches (Chiang, 2005; Simard et al., 2005). Different features have different scopes, as discussed in Section 2.6. Global features are computed from the entire derivation, they include:

- **Distortion count:** Sums the number of source words between two source phrases translated into consecutive target phrases.
- **Phrase penalty:** The number of phrase pairs used in the derivation $|\mathcal{D}|$.
- **Word penalty:** The number of produced target words, which controls the length of translation.

Other features use a limited context around the individual phrase pairs:

- **Target language model:** The logarithm of an n-gram target language model

$$\log p(\mathbf{e}) = \log \prod_{j=1}^{M} p(e_j | e_{j-1} \dots e_{j-n}), \tag{3.3}$$

  which requires to remember a history of n words for each position in the target sentence.

The remaining features require many parameters and can be factorized in terms of individual phrase pairs. They include phrase translation probabilities, lexical weighting and lexical reordering.

- **Translation probabilities:** The conditional translation probability of the target phrase given the source phrase:

$$\log \prod_{(\mathbf{r},\mathbf{p}) \in \mathbf{d}} p(\mathbf{r}|\mathbf{p}), \tag{3.4}$$

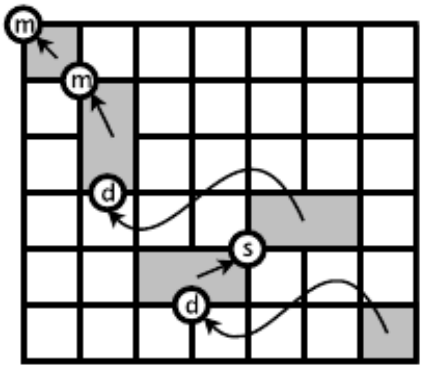


Figure 3.1: *Phrase orientations in a lexicalized reordering model (Koehn, 2010).*

where **p** is a source phrase and **r** is a target phrase. The equivalent phrase probability for the same phrase pairs in the opposite direction $p(\mathbf{p}|\mathbf{r})$ is also used. It has been found to produce a performance comparable to the direct probability $p(\mathbf{p}|\mathbf{r})$ in practice (Och, Tillmann, and Ney, 1999).

The estimation of the individual probabilities vary along with the phrase alignment model used to build the bilexicon. For generative models, these probabilities could correspond to the parameters of the model computed with EM. Alternatively, they could correspond to normalized joint frequencies accumulated over the bilexicon.

$$p(\mathbf{p}|\mathbf{r}) = \frac{\text{count}(\mathbf{p}, \mathbf{r})}{\text{count}(\mathbf{r})} \tag{3.5}$$

The nominator represents the number of the joint occurrences of both phrases aligned together $(\mathbf{p}, \mathbf{r})$, while the denominator represents the marginal counts of the phrase **r**. $p(\mathbf{p}|\mathbf{r})$ is defined similarly.

- **Lexical weighting:** Relative frequency estimation of conditional phrase probabilities are overly optimistic due do data sparsity. Lexical weighting is then basically used as a smoothing method for infrequent phrase pairs, the probabilities of which are poorly estimated (Foster, Kuhn, and Johnson, 2006). Smoothing is based on word-to-word translation probabilities, for which statistics are available. The target-to-source lexical weighting is:

$$\text{lex}(\mathbf{e}|\mathbf{f}, \mathbf{A}) = \log \prod_{j=1}^{M} \frac{1}{|\{i : (i,j) \in \mathbf{A}\}|} \sum_{i:(i,j)\in\mathbf{A}} p(f_i|e_j), \tag{3.6}$$

where **A** refers to some underlying word alignment. The reverse lexical weighting $\text{lex}(\mathbf{f}|\mathbf{e}, \mathbf{A})$ is defined similarly. The word conditional probabilities $p(f_i|e_j)$ are computed in a similar way as phrase conditional probabilities. The parameters of word-based IBM model 1 are found to perform well in practice.

- **Lexicalized reordering:** These features describe the orientation of a source phrase being translated with respect to the previously translated phrase. Reordering can be represented as the distance (in number of words) between these two source phrases. To avoid sparsity issues, *orientation* can be limited to some predefined categories: the most widely used are *monotone*, *swap (s)* with the previously translated source phrase and

Bilexica → ← Features

Source	Target	$p(\bar{e}\|\bar{f})$	$lex(\bar{e}\|\bar{f})$	$p(o_m\|\bar{e},\bar{f})$	...
نصف ما أقوله	I say	0.83	0.12	0.87	...
	what I say	0.11	0.03	0.91	
	half of what I say	0.01	0.01	0.91	
لا معنى له	meaningless	0.91	0.56	0.75	...
	is meaningles	0.09	0.28	1.00	
النصف الآخر	second half	0.27	0.02	0.13	...
	other half	0.36	0.01	0.53	
	the other half	0.14	0.00	0.33	
يبلغك	reach	0.35	0.11	0.45	...
	reach you	0.24	0.02	0.67	

Figure 3.2: *An example phrase table.*

discontinuous (d). These categories are illustrated in Figure 3.1, borrowed from (Koehn, 2010). The associated features are then computed:

$$\log \prod_{(\text{orientation},\mathbf{r},\mathbf{p})\in\mathcal{D}} p(\text{orientation}|\mathbf{p},\mathbf{r}). \tag{3.7}$$

Again, there exist several ways to compute the probabilities $p(\text{orientation}|\mathbf{p},\mathbf{r})$ for all phrase pairs in the bilexicon. A common practice is again to rely on relative frequencies of such events in the parallel corpus annotated with alignment. Orientation events can be defined either with respect to the word alignment (Tillmann, 2004; Koehn et al., 2005) or to the phrase alignment (Galley and Manning, 2008).

Other score functions have been proposed in the literature and can be used in conjunction with the previous scores. In fact, any function associating a numerical positive score with each phrase pair is a candidate feature. Boolean functions can thus be used for measuring arbitrary syntactic properties of a phrase pair, such as "Is $\mathbf{r}$ a target constituent?" "Do $\mathbf{r}$ and $\mathbf{p}$ both contain a verb?", and so on. Additional feature functions relying upon external information such as syntactic parses can also be found in the literature (Och et al., 2004; Chiang, Knight, and Wang, 2009).

3.2.4 The Phrase Table

A data structure that is widely used in phrase-based systems is the *phrase table*. This structure contains all the phrase pairs included in the bilexicon. Since many of the features used by the model are decomposable in terms of individual phrase pairs, they are precomputed and stored in the phrase table as well. Figure 3.2 shows an example of a phrase table. It represents each source phrase along with each possible translation and the associated parameter values. The pipeline used to build the phrase table is pictured in Figure 3.3.

3.2.5 Learning in Discriminative Models

From the definition of the discriminative model defined in Equation (3.2), after computing the values of the features $h_k(\mathbf{e},\mathbf{d},\mathbf{f})_{k=1}^{K}$, the only parameters that remain to be learned are the scaling factors $\lambda_k{}_{k=1}^{K}$.

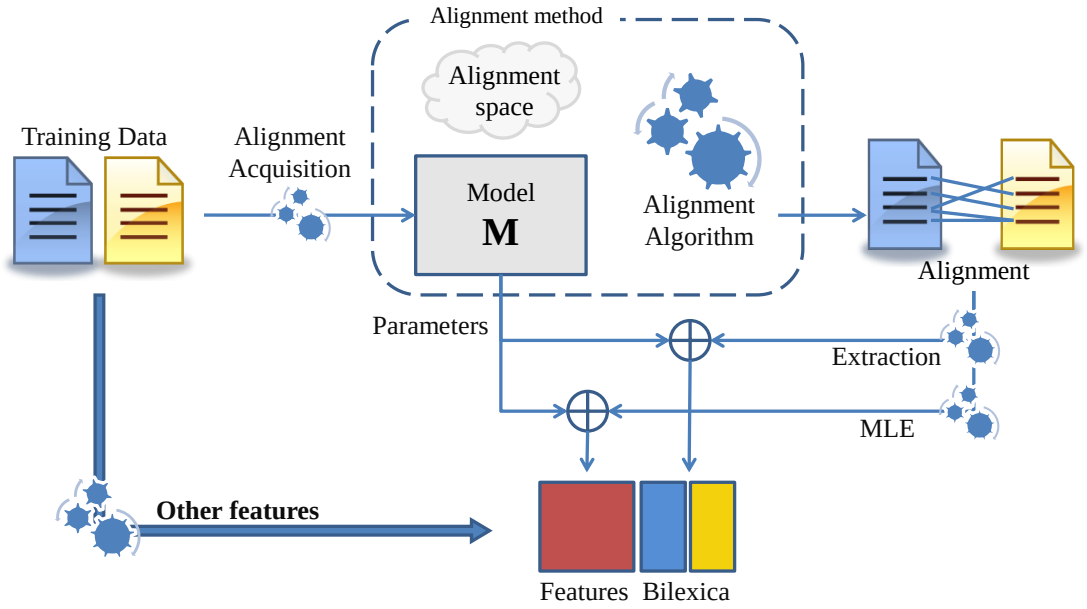


Figure 3.3: *The pipeline to construct the phrase table.*

The MERT algorithm (Och, 2003; Zaidan, 2009) is widely used in practice for the estimation of these parameters. MERT implements Powell search method to find a local optimum of the BLEU function, which is non-differentiable, non-convex, without computing any of its derivatives, This procedure remains widely used in practice in spite of its computational cost[3]; the instability of its solutions due to local minima; and the limitation of the number of weights that can be simultaneously optimized. Multiple variations and improvements are proposed in (Foster and Kuhn, 2007; Cer, Jurafsky, and Manning, 2008; Moore and Quirk, 2008). An alternative method based on large margin approach called MIRA is proposed by (Crammer et al., 2006; Chiang, Marton, and Resnik, 2008). Other approaches use more conventional discriminative learning algorithms (Liang et al., 2006).

3.3 Decoding

Once the model is specified and all the parameters are estimated, it is possible to translate new input sentences. The role of the decoding module is to construct a translation for any source sentence. The best translation hypothesis is the one with the highest model score and therefore translation is a matter of searching, among the sentences which can be aligned with $\mathbf{f}$, the hypothesis $\mathbf{e}^*$ maximizing the linear model score:

$$\mathbf{e}^* = \arg\max_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \arg\max_{\mathbf{e}} \sum_{\mathbf{d}\in\mathcal{D}} p(\mathbf{e},\mathbf{d}|\mathbf{f}) = \arg\max_{\mathbf{e}} \sum_{\mathbf{d}\in\mathcal{D}} \sum_{k=1}^{K} \lambda_k h_k(\mathbf{e},\mathbf{d},\mathbf{f}). \tag{3.8}$$

The search space ranges over $\Lambda^* \times \mathcal{D}$ for a given $\mathbf{f}$ and the optimization involves a sum over exponential number of derivations. Computing the best translation in fact involves the resolution of a NP-hard combinatorial optimization problem when reordering is arbitrary (Knight, 1999).

[3]MERT typically requires multiple decoding of the development set and training a complete system can take several hours, sometimes days to optimize a dozen of parameters

In order to tackle this complexity, one can impose further restrictions on the search space and on the scoring function to allow for efficient resolution strategies. An example would be allowing only monotone or limited local reordering translations. the search algorithm proceeds through a directed acyclic graph of states representing partial or completed translation hypotheses, which are constructed from left-to-right in the target language word order Wang and Waibel (1997); Koehn (2004). Other implementations (Knight and Al-Onaizan, 1998; Kumar and Byrne, 2003; Kumar, Deng, and Byrne, 2006) may rely on the formalism of FSTs, and benefit from well-known and efficient algorithms (Mohri, Pereira, and Riley, 2002). As an alternative way to reduce the complexity of the search, one can resort to heuristic search techniques and compute approximate solutions. Possibilities include the use of *best first* search techniques (Pearl, 1984); greedy *local search* techniques (Germann, 2003); monotone decoding applied on a large permutation sets computed heuristically (Crego and Mariño, 2006); the transformation of the decoding problem into a known combinatorial problem which can then be solved using general purpose solvers such as Integer Linear Programming (ILP) (Germann et al., 2001); etc. To reduce the complexity, these approaches actually drop the marginalization of the derivation (the sum in the Equation 3.8) and search for the best $(\mathbf{e}, \mathbf{d})$[4]. An alternative decision rule is the MBR decoding proposed in (Kumar and Byrne, 2004) which aims at a direct minimization of the expected risk of translation errors under a given loss function such as BLEU.

Even if there are no search errors and the translation that exactly optimizes the decision rule can be produced, the output of the decoder may not be the actual best translation according to human judgment. It is possible that the search space explored by the decoder contained a better translation, and the decoder assigned a lower score for this hypothesis because its score estimation was incorrect. This is called *model error*. One approach to reducing model error is *reranking* or *rescoring* in which the decoder returns N highest-scoring translations for some value N. These translations are then rescored by an alternative model with access to more feature functions than the decoder. This can be done using a log-linear model as in (Och et al., 2004; Shen, Sarkar, and Och, 2004) or any other machine learning approaches such as kernel methods or Gaussian mixture models (Nguyen, Mahajan, and He, 2007). A wide range of features can be useful to improve the re-ranking performance (Giménez and Màrquez, 2008; Chiang, Knight, and Wang, 2009).

Figure 3.4 represents the main components of a log-linear SMT system.

3.4 Evaluating Machine Translation

One way of evaluating the output of an SMT system relies on a comparison between the system's output and correct translations. However, as argued in Section 1.2 translation is non-deterministic. Furthermore, comparison between translations is not well-defined which make judging the quality of one translation with respect to the references a difficult task. The problem of evaluation is usually solved either by asking a human expert to subjectively judge the quality of the system's output; or by explicitly constructing the the correct answer and conceiving an objective comparison metric.

Subjective evaluation requires the annotators to judge the quality of a translation based on several criteria such as intelligibility, fluency, fidelity, adequacy and even informativity. This approach is adopted in recent evaluation campaigns (Callison-Burch et al., 2008; Callison-Burch et al., 2009). Alternatively, the judgment may be based on how helpful the system's output was to the annotator to complete a specific task (Blanchon and Boitet, 2007); or how easy was post-editing the output to obtain a correct translation (Specia, 2011).

[4]In FST terminology, the exact optimization in Equation 3.8 corresponds to a determinization of the FST followed by a shortest path algorithm. The determinization being very costly is dropped in the approximation.

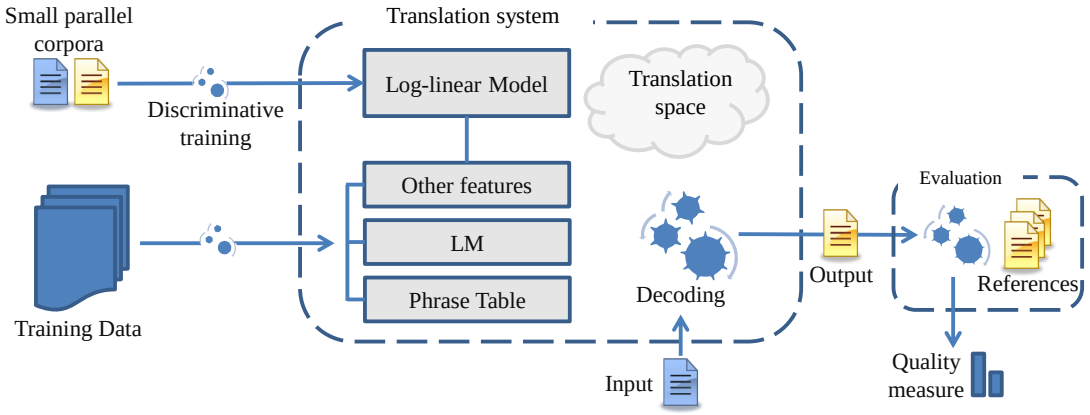


Figure 3.4: *Components of an SMT system. The phrase table, the language model and the other features are first estimated from the training data; discriminative training (MERT, MIRA, etc.) is carried on to learn feature weights from a small development data set. Given all the parameters and an input sentence, the decoder explores the search space and output the best translation.*

Automatic evaluation mostly relies on a direct comparison between the system output hypothesis and the reference translations. The underlying assumption is that the closer the hypothesis is to the reference, the better its quality will be. In comparison with subjective evaluations, human annotator are involved just once in the process, when the reference is generated. The difficulty of automatic evaluation is two-fold. On the one hand, we have the difficulty of defining the correct translation. Usually one or several human experts are asked to translate the input sentence and build the set of references as an approximation of the space of correct translations. However, given the nature of translation this space is huge, and few translations are likely to cover only a small fraction of it. Recent technologies based on *meaning-equivalent semantics* tools (Dreyer and Marcu, 2012) provide the annotators with efficient ways to generate a large number of reference translations,thus resulting in a better approximation of the correct translations space.

On the other hand, there is the difficulty of designing metrics capable of taking into account many aspects of the comparison such as the similarity of syntactic structure or the similarity of semantic content. Current metrics are far form perfect and improving them is still an active research area[5]. The most widely used metric is the BLEU score (Papineni et al., 2002). BLEU considers not only single word matches between the output and the reference sentence, but also n-gram matches, up to some maximum n. This formulation permits to reward sentences where local word order is closer to the local word order in the reference. BLEU is a precision-oriented metric; that is, it considers the number of n-gram matches as a fraction of the number of total n-grams in the output sentence. Other metrics such as TERp (Snover et al., 2006b) and METEOR (Agarwal and Lavie, 2008) have been developed recently, and are becoming more and more popular.

3.5 Summary

SMT is the main application which drives most of the research in alignment models. In this chapter, we have described a state-of-the-art phrase-based SMT system. In the following part, we will use such a system to evaluate the performance of our alignment models.

[5]See the WMT metrics tasks between 2008 and 2012 `http://www.statmt.org/wmt12/`

After having discussed the motivation for using phrases, instead of words, as the units of translation, we have presented the discriminative phrase-based model that we use in this dissertation. This model is a weighted linear combination of feature functions. Translation hypotheses are constructed by concatenating phrase translations found in the bilexicon of the translation system. This bilexicon is typically built from a parallel corpus which is annotated with generalized phrase alignment. We have then detailed the "standard" set of features that are found to be useful and are used in current SMT systems. We have also described the phrase table, which is a data structure, used to store the values of the features for each phrase pair. After having mentioned methods to learn the parameters of this model, we have briefly introduce the inference in this model which amounts to the search for the best scoring translation hypotheses. This search is done by the decoder. Finally, we have presented several automatic metrics for the evaluation of the translation quality.

Part II

Improving Alignment with Discriminative Learning Techniques for Statistical Machine Translation

Research Statement

I have introduced the problem of bitext alignment and its applications, especially SMT, and surveyed the literature for existing approaches. In our research contribution we are interested in improving the intrinsic quality of bitext alignment and its impact on the SMT application. We explore the pipeline of constructing a phrase table from a parallel corpus, spot its weaknesses, and propose several methods to improve it. As shown in Figure 3.5, phrase tables are built from phrase alignments, which in their turn rely on word alignment information.

The first problem we are concerned with is improving the word alignment quality. In Chapter 2 we reviewed state-of-the-art approaches and pointed out their problems. Now we summarize these problems and our propositions to confront them.

- **Asymmetry.** This problem results from representing the alignment as a mapping function from one side of the bitext to the other. Therefore the output depends on the direction of the alignment. Asymmetry limits the alignment to one-to-many patterns, whereas we are interested in many-to-many word alignments. This problem exists in many generative approaches including the widely used IBM models, as well as discriminative approaches. This problem is solved in practice by using a symmetrization heuristic, which starts from the intersection of two directional alignments, and heuristically adds points from their union to increase coverage.

  We propose in Chapter 4 an alternative representation of the alignment by directly modeling the alignment matrix. We propose to model the decision of aligning any word pair using a MaxEnt model. This results in symmetric, many-to-many alignments. Our proposition is also a model-based replacement of the symmetrization heuristic.

- **Incorporating features.** Generative models have to factorize according to a particular generative process, which imposes considerable restrictions on the kinds of features that can be incorporated.

  We propose to use a discriminative approach which facilitates the incorporation of many relevant features.

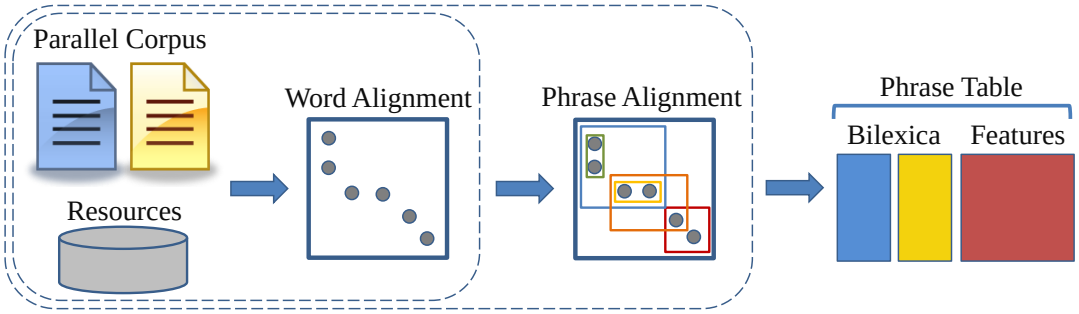


Figure 3.5: *Constructing a phrase table from a parallel corpus.*

- **Structure and independence assumptions.** In both generative and discriminative models there is a compromise about the model structure. A complicated structure, with many dependencies, enhances the model capacity to take the context into consideration when making alignment decisions. However, efficiency and the increased number of parameters become issues for inference and learning. Typically strong independence assumptions are made which do not always correspond to reality.

  The model we propose in Chapter 4 enables efficient learning and inference by making the alignment decision of each word pair independently from the other pairs. Better structure and more dependencies are then incorporated by using stacking machine learning techniques without increasing the model complexity.

- **Estimating link posteriors.** Computing a score to evaluate the association between two words under the alignment model is of a great importance for some alignment methods and for applications of the alignments. This score corresponds to the link posterior probability. Computing posteriors involve summing over all possible alignments, which is intractable under complicated model such as the IBM models.

  Our approach is an efficient way to compute such scores since it models the posterior directly.

- **Correlation between AER and BLEU.** The correlation between intrinsic alignment quality measures, such as AER, and translation quality measures, such as BLEU, has seen contradictory results in the literature. Furthermore, it is not clear what characteristics of word alignments are required in order to produce good translation performance.

  We propose a series of experiments which involve several discriminative and generative alignment models and we compare their characteristics in light of their translation performance.

The second problem we are concerned with is phrase pairs extraction. In the standard approach to phrase-based translation systems, the word alignment is first computed for each parallel sentence; then an extraction heuristic is used to compute the extraction set (or generalized phrase alignment), from which the phrase table is built. We call this approach "Viterbi-based" because it relies on the one-best alignment. We aim to confront the following problems in this pipeline.

- **Alignment error propagation.** Since the standard phrase extraction procedure ignores alignment posterior probabilities, it tends to be sensitive to alignment's precision and recall errors. An erroneous link, as unlikely as it may be, can prevent the extraction of many plausible phrase pairs. Furthermore, the extracted phrase pairs are all considered of equal quality, regardless of how much evidence the alignment matrix provides for them.

  Our MaxEnt model presented in Chapter 4 allows efficient and reliable estimation of the posterior probabilities. We take advantage of this formulation in Chapter 5 and use alternative extraction methods that allow to consider the entire alignment distribution to make extraction decisions. We call this approach "posterior-based" extraction.

- **Balancing precision and recall.** In the standard pipeline, the number of phrase pairs extracted per sentence pair, and hence the size of the phrase table, is determined by the underlying word alignment. However, if large training corpora are available for the SMT system, only precise phrase pairs with high translation quality are needed to be extracted from each sentence; while for smaller corpora, more phrase pairs may be better even of lower quality. Controlling this balance between precision and recall for building phrase tables is not possible in the standard approach.

We propose to remedy this problem in two ways. First, use thresholding of the link posterior probabilities to produce the word alignment. This threshold allows to control the sparsity of the resulting alignment and hence the number of extracted phrase pairs. Second, do not use a single word alignment, but instead use an extraction procedure that use the posteriors directly to compute a confidence score per phrase pair and threshold this later score.

The third problem we are concerned with is the generalized phrase alignment for SMT. Building the phrase table requires the extraction of all relevant phrase pairs. We address the following problems in the existing methods.

- **Incorporating features.** "Posterior-based" extraction approach improves over the "Viterbi-based" by using links posteriors instead of links in the one-best. However, only the alignment models are used which are not perfect.

  In Chapter 6 we propose to use several features in addition to the alignment models in order to recover from their errors.

- **Modeling the extraction.** The predominant methods to build the extraction set (the generalized phrase alignment) for a sentence pair is using extraction heuristics. The extraction decisions are based on the estimation of the intrinsic quality of phrase pairs, estimated from word alignment models or phrase bisegmentation models. However, these heuristics are not concerned with the translation as the final application of the phrase pairs, and are agnostic about the "utility" of extracted phrase pairs in that context.

  In Chapter 6 we present a model-based, discriminative approach to the extraction problem. In our model, extraction decisions are learned from phrase pairs annotated as useful for translation.

Chapter 4 presents a word-based alignment model which estimates the link posteriors directly in a MaxEnt framework. Chapter 5 studies the performance of this model when used in SMT systems, and compare Viterbi-based and posterior-based extraction heuristics. Chapter 6 introduces a novel discriminative extraction model for phrase pairs, which is informed about the quality of the translation.

CHAPTER 4

A Maximum Entropy Framework for Word-Based Alignment Models

The word alignment task is at the heart of many applications including machine translation. They constitute the first step in the process of building the bilexicon in phrase-based SMT systems, as well as in syntax-based systems. Different approaches to solve the word alignment problem have been discussed in Chapter 2. The most widely used in practice are the generative IBM models (Brown et al., 1993). Such models can be easily trained from sentence-aligned bitexts in an unsupervised way using the EM algorithm. Unfortunately, IBM models make a lot of alignment errors, and our main objective in this chapter is to design a word alignment model that improves the intrinsic quality of state of the art alignments.

The major problems with IBM models is their asymmetry. Only directional one-to-many mapping can be obtained from these models which does not reflect the symmetric nature of the alignments. A practical solution is to construct two directional alignments and to symmetrize them in a post-processing step. However, the wide-spread symmetrization heuristic (Koehn, Och, and Marcu, 2003) acts locally at the sentence-pair level and lacks a global view of the entire training corpus. An additional issue is that incorporating features into generative models is difficult.

A natural remedy to these problems is to use discriminative models, trained in a supervised way from parallel corpora annotated with manual alignments. Discriminative models are able to consider arbitrary, possibly overlapping, features. In this context, we cast the alignment task as a classification problem: a binary classifier predicts, for each possible link, whether it should be included or not in the alignment (Ayan and Dorr, 2006a). This discriminative framework models directly the posterior probability of alignment links, thus enabling the use of MBR decoding with a threshold and a specific loss function such as AER, similar to (Kumar and Byrne, 2002).

This approach can be seen as a model-based alignment combination method and a replacement of the symmetrization heuristic used with the generative models (cf. Section 2.3.1). The alignments to be combined are used to compute features and restrict the set of possible links that are passed to the classifier. Combination decisions are learned in light of a global view of the data to maximize the AER, instead of being made locally and arbitrarily as in the heuristic.

However, this approach remains unable to model interactions between alignment decisions

which are of great help to correctly prevent or promote certain configurations in the predicted alignment. For instance, when predicting whether two words are aligned or not, the binary model does not have access to information about the alignment predictions of the neighboring words. This shortcoming may be overcome by introducing a stacked classification layer (Wolpert, 1992) that operates globally on the alignment matrix level and, hence, enables arbitrary features describing interactions between alignment decisions to be taken into consideration.

The main contribution of this chapter is a simple and efficient MaxEnt alignment model, which can be trained from a small amount of labeled data. The model dispenses with the symmetrization heuristic and delivers state-of-the-art alignment quality as measured by AER.

This chapter is organized as follows. In Sections 4.1 and 4.2, we propose to model the distribution over binary alignment decisions with a MaxEnt model. Using this model, thresholding can be used to produce alignments as described in Section 4.3. In Sections 4.4 and 4.5 we describe how we estimate the parameters of our model and how we cope with the problem of imbalanced training data sets. One important aspect in discriminative modeling is the choice of features which we detail in Section 4.6. The model is enhanced with a stacked classification layer described in Section 4.7. In the experiments reported in Sections 4.8 and 4.9 we evaluate the intrinsic quality of the alignments produced by our framework as measured by the AER. We compare our model to other state-of-the-art generative and discriminative models. We also extensively study the role of each component in the model and its relation to the alignment quality. We conclude with in a summary of the chapter in Section 4.11. The findings of this chapter was originally published in (Tomeh et al., 2010; Tomeh, Allauzen, and Yvon, 2011a; Tomeh, Allauzen, and Yvon, 2011b; Tomeh et al., 2011a).

4.1 Word Alignment as a Structured Prediction Problem

The task of word alignment is to find many-to-many word-level, translational equivalences between two parallel sentences $\mathbf{f}$ and $\mathbf{e}$, of length N and M respectively. The alignment refers to the set of links pairing single word *positions* in the two sentences.

Let $\mathcal{N} = \{j : 1 \leqslant j \leqslant N\}$ be the set of source positions and $\mathcal{M} = \{i : 1 \leqslant i \leqslant M\}$ be the set of target positions. A word alignment is defined as:

$$\mathbf{A} = \{(i, j) \in \mathcal{M} \times \mathcal{N}\}. \tag{4.1}$$

Matching is only possible between word positions, meaning that only single words can be *explicitly* put in a translation relation. This can be interpreted as fixed segmentation constraints on the sentences.

This alignment is usually represented by enumerating the function $A : \mathcal{N} \times \mathcal{M} \to \{-1, 1\}$ which maps the cells (i, j) in the *alignment matrix* to a binary value $A_{i,j}$ indicating whether the corresponding words are aligned or not: 1 indicates an *active* link and -1 an *inactive* link. Word alignment can then be seen as a binary classification task, in which the goal is to predict a class $y \in \{-1, 1\}$ for every candidate link in the matrix.

4.2 The Maximum Entropy Framework

In probabilistic modeling, predicting an output y from an input $\mathbf{x}$ is based on the conditional probability distribution $p(y|\mathbf{x})$ which is modeled directly in discriminative approaches. Let $\mathbf{x}$ refer to all input information extracted from the context of the parallel sentence and any annotation thereof. The maximum entropy model of this distribution relies on a generalized log-linear parameterization:

$$p(y|\mathbf{x}) = \frac{\exp \boldsymbol{\theta}^\top \mathbf{g}(\mathbf{x}, y)}{\sum_{\acute{y} \in \{-1,1\}} \exp \boldsymbol{\theta}^\top \mathbf{g}(\mathbf{x}, \acute{y})} \tag{4.2}$$

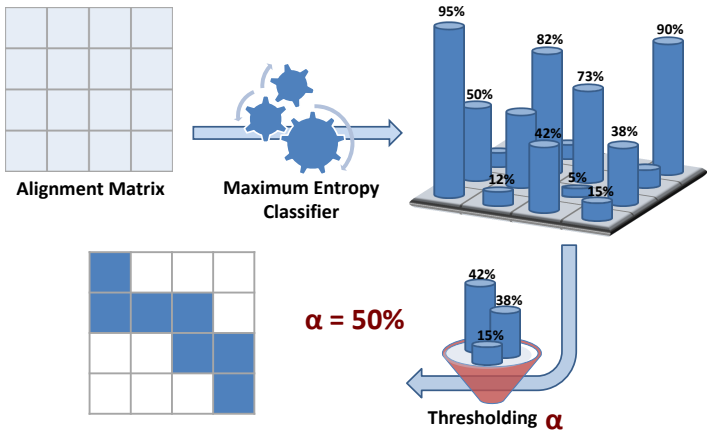


Figure 4.1: *The MaxEnt alignment framework. The classifier is used to populate the weighted alignment matrix. Then, a threshold α is used to select active links.*

The output search space contains two elements $y = A_{i,j} \in \{-1, 1\}$. The partition function $Z(\mathbf{x}) = \sum_{\acute{y} \in \{-1,1\}} \exp \boldsymbol{\theta}^\top \mathbf{g}(\mathbf{x}, \acute{y})$ is specific to each input $\mathbf{x}$ and is used as a normalizer. $\mathbf{g}$ is a feature vector of K components, each of which is associated with a model parameter (a component of the vector $\boldsymbol{\theta}$). Since $Z(\mathbf{x})$ does not depend on y, the decoding rule becomes: $\hat{y} = \arg\max_{y \in \{-1,1\}} \boldsymbol{\theta}^\top \mathbf{g}(\mathbf{x}, y)$. The binary classification case is also called *logistic regression*.

4.3 Minimum Bayes-Risk Decoding

During inference, the model assigns a probability to each possible alignment link. The final output matrix consists of active links whose probability exceeds a threshold ρ (optimized on a development set using grid search). This parameter is used to control the density of the resulting alignment.

Thresholding the link posterior probability $p(a_{j,i}|\mathbf{x})$, which we model directly using the MaxEnt model, is equivalent to the MBR decoding with the AER as loss function (Kumar and Byrne, 2002). Irrespective of the particular choice of the loss function, the threshold allows to trade-off precision and recall. This is shown in Figure 4.1. This decoding results in symmetric many-to-many alignments.

4.4 Parameter Estimation

Given an annotated corpus $\{(\tilde{\mathbf{e}}_k, \tilde{\mathbf{f}}_k, \tilde{\mathbf{A}}_k)\}_{k=1}^{\tilde{N}}$, the conditional log-likelihood function is given as:

$$\Phi_{LL}(\boldsymbol{\theta}) = \frac{1}{\tilde{N}} \sum_{k=1}^{\tilde{N}} \boldsymbol{\theta}^\top \mathbf{g}(\tilde{\mathbf{x}}_k, \tilde{\mathbf{y}}_k) - \log Z_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_k) \tag{4.3}$$

The model is trained to optimize the log-likelihood:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \Phi_{LL}. \tag{4.4}$$

Training is sketched in Figure 4.2. Log-linear models are also called *maximum entropy* (MaxEnt) models, because they can be alternatively derived by maximizing entropy subject to some empirical constraints (Berger, Pietra, and Pietra, 1996).

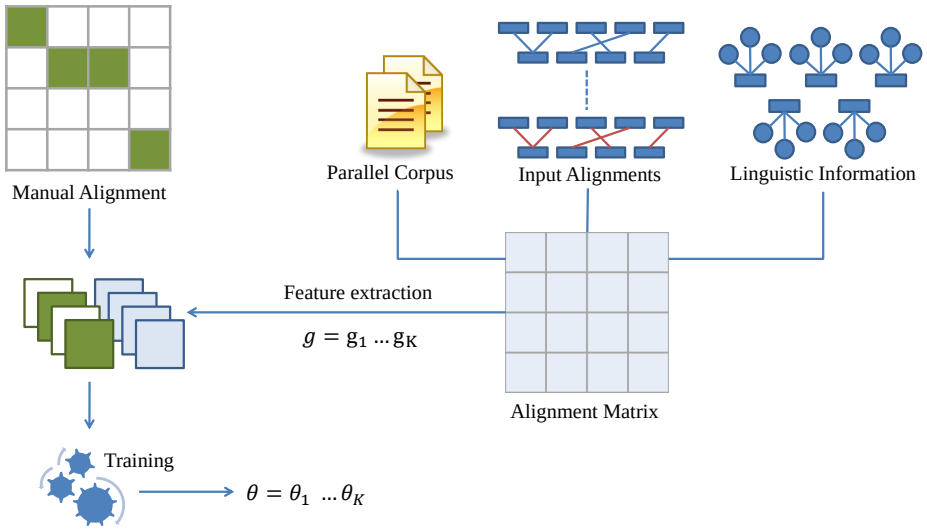


Figure 4.2: *Parameter estimation for the MaxEnt alignment model. The threshold α is optimized separately on a development corpus to maximize the AER.*

MLE for conditional log-linear models does not have a closed-form solution. However, it yields an unconstrained optimization problem with a smooth, differentiable and globally concave function. Therefore, a wide range of numerical optimization algorithms are available to perform the optimization. In most cases, those algorithms will require the calculation of the objective function Φ_{LL} and of its first derivatives with respect to each component θ_i:

$$\frac{\partial \Phi_{LL}}{\partial \theta_i}(\boldsymbol{\theta}) = \mathbb{E}_{\tilde{p}(\mathbf{x},\mathbf{y})}[g_i(\mathbf{x},\mathbf{y})] - \mathbb{E}_{\tilde{p}(\mathbf{x}).p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})}[g_i(\mathbf{x},\mathbf{y})] \tag{4.5}$$

where g_i is the ith feature function. The first derivative with respect to the ith weight is the difference of the expectations $\mathbb{E}$ of the ith feature respectively with respect to the empirical and the to model distributions.

In order to avoid overfitting in MLE, the model is trained to optimize the regularized log-likelihood of the parameters. The most common regularization used in literature is the Gaussian prior (ℓ^2 penalty) which reduces overfitting and thus improves performance on most tasks. An alternative is to use a Laplacian prior (or ℓ^1 penalty). Such regularizer performs feature selection and yields sparse parameter vectors (Tibshirani, 1996). The regularization hyper-parameter aims to control the strength of the regularization.

This optimization requires a fully derivable function to optimize, which is not the case at zero for the ℓ^1 penalty. To overcome this problem, an adaptation of the classical L-BFGS, called OWL-QN (Andrew and Gao, 2007) can be used. In addition to the ℓ^1 regularization term, a small ℓ^2 term is also added to overtake numerical problems that can results from using the second order method, leading to the so called *elastic-net* penalty (Zou and Hastie, 2005). The benefits of the *elastic-net* regularization are two-fold. It enables efficient features selection, without any loss in resulting model quality. Moreover, the obtained models are interpretable, thus enabling to analyze the features contribution. It should be noted that these advantages do not entail a change in the number of model parameters, nor a higher computational cost.

4.5 The Set of Input Links

Since the alignment matrix is typically sparse, with a majority of inactive links, the classification task we consider is imbalanced. Whenever a class is over-represented, its prior

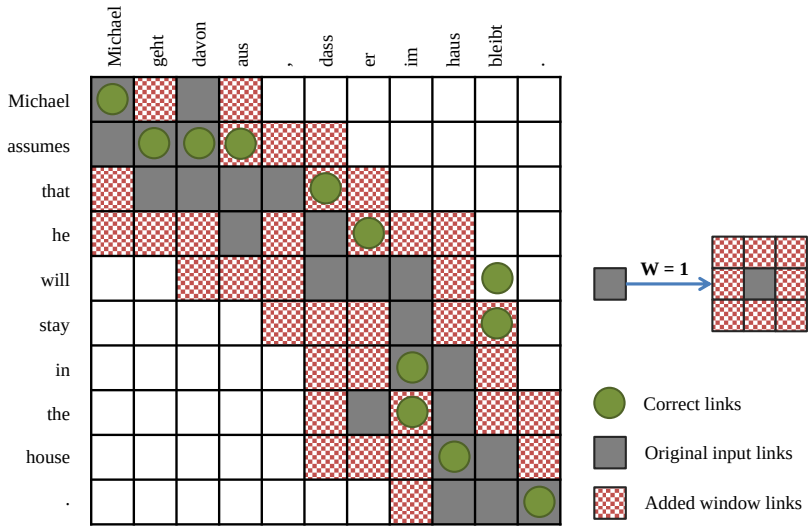


Figure 4.3: *Using a window-based heuristic to extend the input links set.*

probability is higher than that of under-represented classes. Hence, attention should be payed to avoid learning a biased classifier with a tendency towards labeling all links as inactive. For this purpose, we do not consider the entire alignment matrix: we use the input alignments to select a set of permissible links, hoping to obtain a more balanced dataset.

Therefore, the union of all input alignments is used, to select input links. The same method is used during inference, reducing the number of links to be predicted to a subset of the alignment matrix (Ayan and Dorr, 2006a; Habash and Sadat, 2006): only points that have been proposed by at least one input alignment are labeled by the classifier, the others are assumed to be *inactive*.

This reduction of the number of input links (links considered by the classifier) implies an upper bound on the recall, by excluding a lot of plausible links, which then become unreachable by the inference strategy. While a perfect precision can be achieved, recall becomes a bottleneck. The practical effect of the pruning method on the best obtainable alignment (oracle) is studied in details in later sections.

As in alignment combination heuristics, the union of all the input alignments[1] is used to restrict the set of input links. This establishes an upper bound on recall, which can be enhanced by adding the links in a neighborhood defined using a fixed-size window strategy. This method is motivated by the observation that good candidate alignment points often neighbor other good alignment points. Figure 4.3 illustrates the extension of the set of input links using the window-based heuristic. A down side for this heuristic is the increased number of negative examples, which may contribute to the imbalanced data problem[2]. Possible solutions include random sub-sampling of the training data, and adjusting the selection threshold to neutralize the a priori probability assigned to the over-represented *inactive* class.

4.6 Features

The choice of features is critical for the performance of a model, and many research questions involve primarily the exploration of new features for a particular task. In NLP, feature

[1] For instance, IBM models can be used to generate input alignments.
[2] In practice, we do not observe such degradation in performance

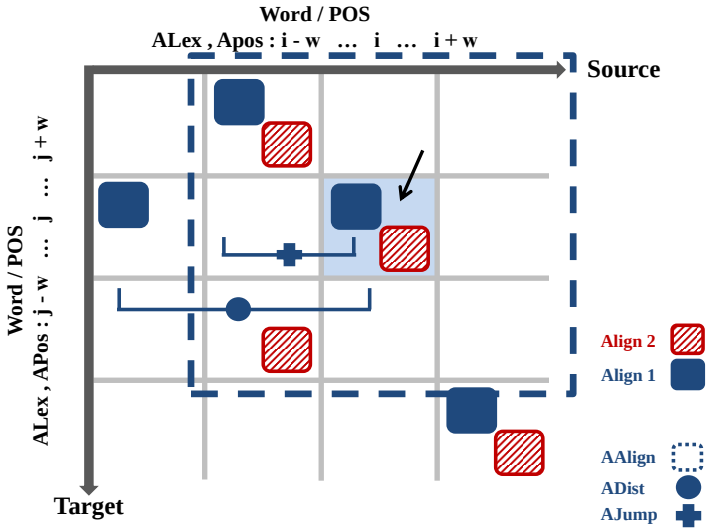


Figure 4.4: *Features extracted to label the link pointed to by the arrow. Align1 and Align2 are input alignments, AAlign is the window that defines the context from which the input alignment based features are extracted. ADist and AJump show the value of the respective features for this specific matrix.*

engineering is largely a matter of manual development guided by linguistic expertise and task performance. The main strategy is to incorporate as many features as possible into learning and to allow the parameter estimation method to determine which features are helpful and which should be dismissed. However, caution should be taken. Adding more features can only help a model fit the training data better, but at the risk of overfitting, with negative effects on performance on new data. Overfitting can be reduced using ℓ^1 regularization, as described earlier in Section 4.4. Discretization of continuous features is performed in a preprocessing step, using an unsupervised equal frequency interval binning method (Dougherty, Kohavi, and Sahami, 1995).

In our discriminative model, we consider two kinds of features: word and alignment matrix features; some of them are illustrated in Figure 4.4.

4.6.1 Word Features

Word features aim to describe the linguistic context of a given link, and depend on the sentence-pair in which it occurs, augmented by part-of-speech tags and related corpus statistics. They include:

1. Part-of-speech tags (**WPOS**) for a window of words, with variable size, surrounding the source and target words. This window size variable introduces a model parameter to be used to optimize AER on a development set. An example of this feature could be:

$$g_{\text{WPOS}}(i, j, \mathbf{e}, \mathbf{f}) = \begin{cases} 1 & \text{if POS}(e_j) = \texttt{VERB} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

   The WPOS feature is computed for all target words at positions $j-2, j-1, j, j+1, j+2$, and similarly for source words. These features help to capture syntactic patterns in the alignment.

2. Surface lexical form (**WLex**), which is active if the source/target word is one of the L most frequent words. Again, L introduces an additional hyper-parameter. These

features help aligning frequent words by boosting the weight of their correct associations (encountered in the manual alignments).

3. Monotonicity (**WMono**) of the link $a_{j,i}$ which includes the difference between source and target absolute positions $|i-j|$ and their relative positions to the sentence length $\frac{i}{M}$, $\frac{j}{N}$ and $|\frac{i}{M}-\frac{j}{N}|$. These features capture the distortion information by computing the (normalized) distance from the diagonal of the matrix.

4. Lexical probability (**WProb**). These features include a separate feature for each discretized probability $p(f_i|e_j)$ and $p(e_j|f_i)$. We use the parameter of IBM model 1 to compute these features.

5. Word frequency (**WFreq**). The source and target word frequency (and their ratio) computed as the number of occurrences of the word form in the training data.

6. Lexical Prefix/Suffix (**WPref**,**WSuff**) A separate feature for each prefix/suffix of a predefined length (and their combination), for $a_{i,j}$ source and target words. An example of a combination feature:

$$g_{PS}(i,j,\mathbf{e},\mathbf{f}) = \begin{cases} 1 & \text{if prefix}(\mathbf{f}_i) = \texttt{Al} \wedge \text{suffix}(\mathbf{f}_i) = \texttt{At} \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

7. Word similarity (**WSim**). These features reflect that proper nouns are often spelled similarly in different languages, e.g. "SdAm Hsyn" [3] and "Saddam Hussein". A separate feature is defined per distinct value of the word similarity between f_i and e_j. We use the Levenshtein (edit) distance as a measure of similarity.

8. Identity (**WIdent**), which is active whenever f_i is equal to e_j, which can be useful for untranslated numbers, symbols, names, and punctuations.

9. Punctuation mismatch **WPunct**. These features indicate whenever a punctuation is aligned to a non-punctuation.

Any number of other information sources can be used to design additional lexical features such as word classes, chunks, stems, parse trees, etc.

4.6.2 Alignment Matrix Features

These features characterize the set of input alignment matrices, in addition to their union matrix $\mathbf{A}_\cup$. Most of these features have been already proposed (Ayan and Dorr, 2006a; Blunsom and Cohn, 2006; Habash and Sadat, 2006), exceptions are ADist and AJump, which are novelties of this work. Our feature set includes:

1. Predictions (**AAlign**) of individual input alignment systems (and their union $\mathbf{A}_\cup$) for the current link and its neighborhood in a window of size $w \times w$ where w is an additional model meta parameter.

   These features test whether a particular link exists in this neighborhood according to each input alignment. We also test the total number of input alignments supporting it. Neighbor features are used to inform the current link about its surrounding points, motivated by the fact that alignment points are usually found around the diagonal of the alignment matrix.

[3] All Arabic transliterations are provided in the Buckwalter transliteration scheme

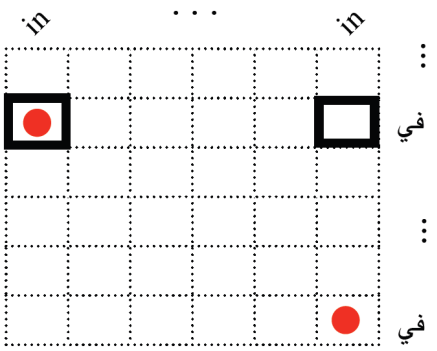


Figure 4.5: *A common problem with IBM Model 4 alignments is a too weak distortion model. The second English "in" is aligned to the wrong Arabic token. Circles show the gold alignment.*

2. Source and target word fertility (**AFert**), which represent the number of target (source) words aligned to the current source (target) word according to a given input alignment and/or to the union alignment;

3. Distance features (**ADist**), which describe the minimum/maximum distance between the current link and the previous/following links of same line/column according to the union alignment matrix. Beside characterizing fertility and monotonicity of the union alignment, distance features provide information about the bi-phrases that can be extracted from the alignment. The larger the distance, the fewer the extracted phrases and the more discontinuous they are;

4. Jump features (**AJump**), which characterize the absolute distance between the current word and closest aligned one, on both source and target side according to the union alignment matrix. These features provide information about gaps in the alignment.

5. Multiple distortion (**AMultd**) features, which indicate whether a link involves a duplicated word. Indeed, duplicated words are often misaligned due to a weak distortion model in comparison with lexical probabilities in IBM alignments (Riesa and Marcu, 2010). E.g. several "fy" on the source side could be erroneously aligned to the same "in" on the target side regardless of the distortion. This feature is active for the link $a_{i,j}$ if f_i or e_j is duplicated, returning the distance to the diagonal. Figure 4.5, borrowed from (Riesa and Marcu, 2010) illustrates the utility of such feature.

4.6.3 Partitioning Features

Following (Ayan and Dorr, 2006a; Blunsom and Cohn, 2006), each feature function is conditioned twice on the POS tags of the source word and the target word. We also add another conditioning criterion corresponding to their conjunction. Thus, we learn a separate weight for each feature for each source, target and source/target POS tags, allowing the model to pay more or less attention to each feature depending on the related tags.

4.7 Stacked Inference

Two issues with the MaxEnt formulation of the alignment problem are that i) structure is not taken into account; and ii) labels are predicted independently. While this keeps the model simple, interactions between individual predictions can not be modeled.

One can solve this problem by predicting the entire alignment matrix at once using, for instance, multinomial logistic regression, conditional random fields, large-margin based method, or any other structured prediction approach (cf. Section 2.3.4). However, models with a lot of dependencies are difficult to learn and are not always tractable. In order to incorporate structure and dependencies into the MaxEnt model, without sacrificing efficiency, we use a *stacked inference* method (Wolpert, 1992).

Stacked inference is merely an approximation to structured learning. It allows us to indirectly model dependencies between predicted labels at a low computational cost. It has been successfully applied to several NLP problems, like dependency parsing (Martins et al., 2008), named entity recognition (Krishnan and Manning, 2006) and sequential partitioning problems (Cohen and Carvalho, 2005).

4.7.1 The Stacking Algorithm

In stacked learning, all labels are predicted in two steps.

1. For each training example $(\tilde{\mathbf{x}}_k, \tilde{\mathbf{y}}_k)$, the entire set of observations $\tilde{\mathbf{x}}_k$ is considered to extract features, which are then fed to a *first-level* classifier. This classifier is used to assign a label y_i to each observation $\tilde{\mathbf{x}}_k$ without taking dependencies between labels into consideration; then

2. observations are augmented with predictions of the local classifier

$$\mathbf{y}_k = (a_{0,0}, \ldots, a_{j,i}, \ldots, a_{N,M}) \tag{4.8}$$

to generate an *extended representation* of the training corpus, on which a *second-level* classifier is trained. This classifier approximates links interactions using the predictions of the first-level classifier.

4.7.2 A K-fold Selection Process

When building training data for the global classifier, a K-*fold* selection process is used. The entire training dataset is divided into K blocks, and K first-level classifiers are trained, each on a different subset (of $K-1$ blocks) of training data. Each of these classifiers is then used to label the held-out block. These predictions, along with the original data, constitute the training examples for the second-level classifier.

Stacking avoids the explicit joint modeling of labels and is thus merely an approximation method of structured learning. Nevertheless, it allows us to take any type of dependency into account without complicating the model. The runtime of the training algorithm is $O(KT_f + T_s)$ where T_f and T_s are the individual runtimes required for training a first- and a second-level classifier respectively.

4.7.3 Stacking for Word Alignment

For the task of alignment matrix prediction, the use of stacking consists in augmenting input alignments by one additional matrix, which is the output of the first-level classifier.

Over this matrix, features characterizing the interactions between links in the final output alignment can be computed. The same set of features used for the first-level classifier is also used for the second-level one. That is we label the data with a first pass aligner and then we train another model using its prediction as features.

Features like *ADist* and *AJump* are more suitable to capture characteristics of symmetric alignment matrices like the union alignment and the output of the first-level classifier, and hence, are calculated exclusively for them.

Data source		#Sentences	#Ar tokens	#En tokens
IBMAC	*test*	663	16K	19K
	dev	3,486	71K	89K
	train	10K	215K	269K
MT'08	*test set*	1,360	43K	53K
MT'06	*dev set*	1,797	46K	55K
MT'09	*constrained track*	5M	165M	163M

Table 4.1: *Experimental data: number of sentences and running words. The number of tokens after the preprocessing is given.*

4.8 Experimental Methodology

As discussed in Chapter 1, intrinsic and extrinsic methods can be used to evaluate the quality of word alignment. We use AER and several additional measures presented by Guzman, Gao, and Vogel (2009) that characterize the word alignment and the phrase alignment that can extracted from it using the extraction heuristic (Koehn, Och, and Marcu, 2003). We also investigate the relationship between word alignments and the machine translation quality as measured by BLEU when these alignments are used.

4.8.1 Experimental Setup and Data

We experimented the various models with the Arabic-English language pair using the data described in Table 4.1. POS tags for English are generated using the Stanford Tagger[4], while a POS tagger provided by ArabicSVMTools is used for Arabic.

The IBM Arabic-English aligned corpus (IBMAC) (Ittycheriah, Al-Onaizan, and Roukos, 2006) provides gold word alignments used for training and evaluation. It includes a training set that we split into disjoint train and dev sets, used respectively for training and tuning our discriminative models. We use the IBMAC test set (NIST MT Eval'03) to evaluate different alignments in terms of *Alignment Error Rate* (AER).

For MaxEnt training we used freely available toolkits: MaxEnt++[5] and Wapiti[6] (Lavergne, Cappé, and Yvon, 2010).

Basic preprocessing is performed for English. This includes tokenizing punctuations and lowercasing all the tokens, except those recognized as named-entities. We use an in-house tool for named-entity recognition.

4.8.2 Arabic Pre-processing

Arabic is a morphologically complex, highly-inflected language. It has a set of attachable clitics to be distinguished from inflectional features. They are written attached to the word

[4]http://nlp.stanford.edu/software/tagger.shtml
[5]http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html
[6]http://wapiti.limsi.fr

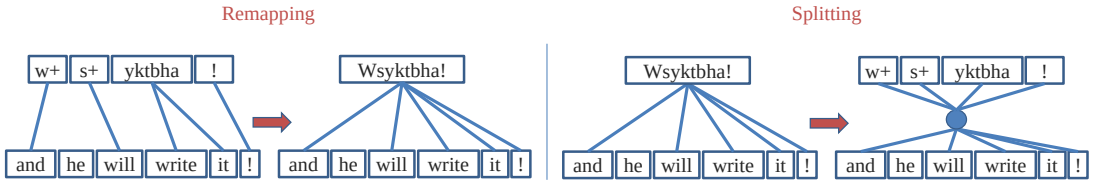


Figure 4.6: *Splitting and remapping of alignments for the Arabic token "wsyktbhA!".*

and thus increase its ambiguity. For example the word form "وسيكتبونها" ("wsyktbwnhA")[7] has two proclitics, one circumfix and one enclitic[8]:

wsyktbwnhA
w+ s+ y+ ktb +wn +hA
and will 3rd person write masculine-plural it
translation: and they will write it

This makes normalization necessary to reduce the sparsity of the data.

We use MADA+TOKAN[9] (Habash, 2007; Roth et al., 2008; N. Habash and Roth, 2009) for morphological analysis, disambiguation and tokenization for Arabic. Given previous experiments on the NIST MT'09 task, we use the *D2* tokenization scheme that showed to perform best under large resource conditions (Habash and Sadat, 2006). For example, the previous Arabic word form "wsyktbwnhA"[7] ("and they will write it" in English) is tokenized according to the D2 scheme as follows: "w+ s+ yktbwnhA".

4.8.3 Remappings Alignments

Since the hand-aligned IBMAC corpus is not tokenized with the MADA+TOKAN D2 scheme, two issues arise:

1. For **evaluation**, the IBMAC manual alignments and the ones estimated on D2-tokenized data should be made compatible. Hence all words need to be mapped back (remapped) to the original form before pre-processing. For an example we consider the Arabic token "wsyktbhA!". An aligner will link the tokens in "w+ s+ yktbhA !" to different words on the English side. In the remapping step, the union of these links is assigned to the original word "wsyktbhA!".

2. For **training**, it is the other way around. The IBMAC manual alignments are split to match the tokenized words. When tokenizing an Arabic word, aligned to some English word(s), all resulting tokens are assumed to have the same set of alignment links as the original word. For instance, suppose that the word "wsyktbhA!" is aligned to all English words in "and he will write it!" in the IBMAC corpus. After applying the D2 tokenization scheme, we link each of the resulting tokens to all the English words. Although this assumption results in noisy reference alignments, it is still the easiest way to obtain reference alignments for D2 tokenized training data.

Figure 4.6 shows an example of splitting and remapping.

[7] All Arabic transliterations are provided in the Buckwalter transliteration scheme
[8] For more details about Arabic processing, we refer the reader to (Habash, 2010)
[9] http://www1.ccls.columbia.edu/MADA/index.html

4.9 Results

This section provides an empirical evaluation of our model by examining the intrinsic quality of the alignments compared to manual alignments. When comparing alignments to a gold standard, the most commonly used metric is the alignment error rate (AER) (Och and Ney, 2003) discussed in Chapter 1. Usually gold alignments are marked with "sure" or "possible" labels, but since the IBMAC corpus has only sure ones, the AER reduces to a balanced $1 - F_\alpha$ measure with $\alpha = 0.5$:

$$Pr = \frac{|A \cap S|}{|A|} \qquad Rc = \frac{|A \cap S|}{|S|} \tag{4.9}$$

$$F_\alpha = \frac{Pr\ Rc}{\alpha Rc + (1-\alpha)Pr} \tag{4.10}$$

where Pr denotes the precision and Rc the recall. We also use F_α with different values for α in the F_α to vary the trade-off between precision and recall as desired: α less (greater) than 0.5 weights recall (precision) higher (Fraser and Marcu, 2007b). Following common practice we do not consider `null` alignment links in the evaluation.

We start by a comparison with other state-of-the-art aligners, including both generative and discriminative models. We then focus on our MaxEnt model and provide a detailed examination of the contribution of each component of the system. First, we study the effect of pruning on the upper bound established on recall and show that stacking makes room for significant improvements. Then we explore the relation between alignment quality and the size of training data and the method of regularization. We show that adding ℓ^1 regularization result in sparser models than ℓ^2 without degrading the performance. We then analyze the obtained models and provide examples of the most useful features which is followed by an extensive evaluation of the contribution of each feature function. We then provide a set or experiments demonstrating the ability of our model to control the balance between precision and recall in order to maximize the AER. Additional experiments to study of the relation between the quality of the input and output alignments are also provided.

4.9.1 Comparison to Generative "Viterbi" Alignments

We compare the MaxEnt alignments to the Viterbi alignments obtained from generative IBM and HMM models. A large scale experiment is conducted using MT'08 training data.

4.9.1.1 Baselines: IBM and HMM models

Table 4.2 summarizes our baseline results obtained with three classical generative alignment models, as estimated by GIZA++[10], in both translation directions, and symmetrized using the *grow-diag-final-and* heuristic (Och and Ney, 2003).

Each step from IBM1 to IBM4 through HMM and IBM3 expectedly results in a better performance. The HMM model achieves a large error reduction over IBM1, with limited added computational complexity. While IBM3 and IBM4 continue to improve the quality of the alignments over HMM, they are much more computationally expensive (learning them takes a few days instead of a few hours) with smaller relative error reduction.

Ar $\rightarrow$ En alignments are always better than En $\rightarrow$ Ar, which is due to differences in morphology between Arabic and English: Arabic is more morphology-rich than English, therefore an Arabic word tend to be translated (and hence aligned) to several English words; in the Ar $\rightarrow$ En direction this one-to-many mapping can be achieved, while it is not possible in the other direction. More aggressive tokenization schemes than D2 should reduce this difference.

[10] http://code.google.com/p/giza-pp/

Model	Direction	Pr%	Rc%	AER%
IBM1	Ar → En	56.4	66.2	39.1
	En → Ar	41.3	64.8	49.6
	GDFA	70.2	71.0	29.4
HMM	Ar → En	66.8	78.4	27.9
	En → Ar	51.0	72.6	40.1
	GDFA	73.9	81.3	22.6
IBM3	Ar → En	68.5	80.4	26.0
	En → Ar	56.5	77.3	34.8
	GDFA	**75.2**	83.8	20.7
IBM4	Ar → En	71.0	83.3	23.3
	En → Ar	58.9	79.8	32.3
	GDFA	75.0	**86.3**	**19.8**

Table 4.2: *AER, precision and recall results for GIZA++ alignments with the GDFA symmetrization heuristic (cf. Section 2.3.1.1).*

Input Alignments [#]	stack?	Pr%	Rc%	AER%
IBM1 [2]	✗	90.4	71.1	20.4
	✓	90.9	72.9	19.6
HMM [2]	✗	90.5	80.7	14.7
	✓	91.0	81.0	14.3
IBM3 [2]	✗	91.1	81.4	14.0
	✓	91.0	81.9	13.8
IBM4 [2]	✗	91.9	83.1	12.7
	✓	92.4	83.0	12.6
IBM1+HMM [4]	✗	91.0	81.7	13.9
	✓	92.9	81.5	13.2
ALL [8]	✗	92.3	84.0	12.1
	✓	92.1	84.4	**11.9**
IBM4 GDFA *baseline*		75.0	86.3	**19.8**

Table 4.3: *Precision, recall and AER results for different sets of input alignments and stacking. [#] is the number of input alignments, "stack?" denotes if stacking was used or not.*

For all the models, the symmetrization heuristic is able to improve both precision and recall, therefore AER, over the combined alignments.

4.9.1.2 MaxEnt and stacking

Table 4.3 reports precision, recall and AER results obtained for the MaxEnt alignments for different set of input alignments. The best alignment with maximum entropy approach, augmented with stacking, achieves a much better precision than the best generative alignment, with slightly worst recall and yields a 11.9% AER (a relative error reduction of 39.9% over the best GIZA++ alignment).

The combination of the four generative models (IBM1, HMM, IBM3, IBM4) yields further improvement with an AER of 12.1% (11.9% with stacking). Stacking systematically improves

the performance and achieves a state-of-the-art AER of 11.9% on the IBMAC test set.

We also note that the difference between the worst precision (90.4%) and the best precision (92.9%) for all MaxEnt alignments, is much smaller than the difference between the worst recall (71.1%) and the best recall (84.4%). This result suggests that the MaxEnt approach easily achieves a good precision even when using noisy input alignments. However, it is more difficult to improve its recall because of the upper bound imposed by the recall of the union of these input alignments. We also note that the IBM4 yields a higher recall than MaxEnt. This result could be explained by observing the number of unaligned words for the two methods: 1422 and 2460 unaligned source and target words respectively for GDFA; and 3371 and 4542 MaxEnt. Manual alignments produce numbers in between the two methods: 2655 and 3457. This is because GDFA iterates over source and target words, trying to leave no word unaligned. This behavior resembles that of human annotators. MaxEnt, however, achieves this indirectly by learning from human annotations, and it is, therefore, less sensitive to unaligned words. More words are left unaligned with MaxEnt than with GDFA, which affects the recall negatively.

The discriminative model systematically outperforms IBM models and the symmetrization heuristic. First, when combining two IBM1 directional alignments, an AER of 20.4% is achieved (19.6% with stacking) which is a big improvement compared to an AER of 29.4%, the result of combining the same two alignments with the symmetrization heuristic (a relative error reduction of 33.3%, when stacking is used). This result is quite impressive since the best input alignment from IBM1 has an AER of 39.0%, which means that even when using noisy input alignments, the MaxEnt model is able to perform a good error correction.

Further more, the MaxEnt model, using only IBM1 alignments, achieves comparable performance with the symmetrization heuristic using IBM4 alignments. This result is interesting since IBM4 is much more computationally expensive than IBM1. Moreover, we can use more accurate input alignments to increase the gain: combining HMM alignments yields a relative reduction of AER of 28%.

4.9.2 Pruning and Oracle Study

As explained in Section 4.5, limiting the set of input links to the union of input alignments, establishes an upper bound on the recall, preventing the model from reaching plausible links. In this oracle study, we quantify manual alignment reachability by several combination of input alignments, with different window sizes.

Table 4.4 displays the percentage of the alignment matrix covered by the union of input alignments, with its recall and AER according to the gold alignment.

Oracle AER drops drastically when increasing the size of the window. Take, for instance, the case of IBM1 models: using a window of size 1 instead of 0 reduces the oracle by 10.8 points (from 13.7 to 2.9) at the cost of exploring a much larger area of alignment matrix (23.5% instead of 4.1%).

It is worth noticing that the HMM model achieves similar oracle scores as IBM4, while its training and inference are fast and exact. Moreover, combining IBM1 and HMM results in performances that are comparable with the standard symmetrization heuristic (which has an oracle of 6.0 for the best IBM model), while exploring a slightly larger set of input links.

Increasing the window size allows us to largely outperform the heuristic with a much larger set of input links. This study suggests that most manual alignments are proposed by the input generative models, and justifies their use to prune the set of input links (an AER of 0.1 can be achieved by exploring 47% of the matrix).

4.9.3 Discriminative Training Set Size

The discriminative approach requires hand-aligned data that are expensive to obtain. Hence, we are interested in knowing how many aligned sentences are needed to train a model that

Input Alignments	Search Space %			Union Recall %			Oracle AER %		
	W=0	*W=1*	*W=2*	*W=0*	*W=1*	*W=2*	*W=0*	*W=1*	*W=2*
IBM1	4.1	23.5	43.9	75.9	94.3	98.7	13.7	2.9	0.7
HMM	3.3	15.9	26.9	85.3	97.0	98.7	7.9	1.5	0.6
IBM4	**3.3**	15.7	26.6	88.7	98.4	99.4	**6.0**	0.8	0.3
IBM1 + HMM	**5.0**	**25.4**	45.4	87.3	98.3	99.6	**6.8**	**0.8**	0.2
ALL	5.5	26.8	**47.0**	90.8	99.2	99.7	4.8	0.4	**0.1**

Table 4.4: *Search space coverage for different window sizes, and associated Oracle AER for different input alignments. W is the window size.*

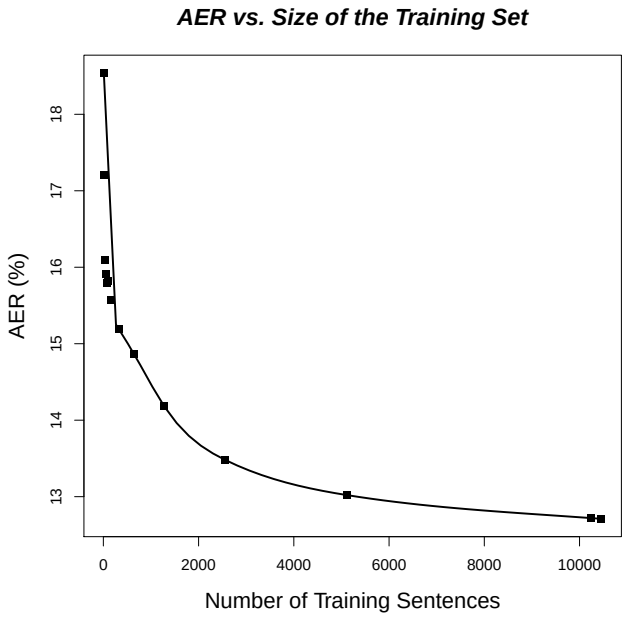


Figure 4.7: *Relation between AER and the number of sentences in the training set when combining two IBM4 alignments.*

performs reasonably well. Figure 4.7 depicts AER as a function of the size of the training set (number of sentences) when using IBM4 as input alignments. Although the bigger the training set the better the model, only small improvements are achievable when using more than 2000 sentences. It is worth noting that with only 10 training sentences (392 training examples), we get an AER of 18.5% which is lower than the AER obtained with the GDFA symmetrization heuristic (19.8%).

4.9.4 Features Analysis

In the previous experiments, presented in Section 4.9.1, we used large-scale systems from all the data we have. These systems uses only a subset of the features we described in Section 4.6. We refer to this subset of features as *Group1*, to distinguish them from the remaining features, which we therefore call *Group2*. In the following experiments, we first analyze the contribution of Group1 features on the large-scale systems, then we use Group1 to build a baseline for Group2 using a smaller amount of data. Table 4.5 enumerates the features in

each group. A subset of Group1 was already used in (Ayan and Dorr, 2006a) and we refer to it as *basic* features.

Group Name	Features
Group1	WPOS, WLex, WMono, AAlign, AFert AJump, ADist
Group2	WPreff, WSuff, WProb, WFreq WIdent, WPunct, WSim, AMultd

Table 4.5: *The two feature groups.*

4.9.4.1 First feature group

To assess the impact of different kinds of features, a contrastive experiment is reported in Table 4.6. The *basic* set of feature families is adapted from (Ayan and Dorr, 2006a). The *Group1* set of features includes three additional families namely ADist, AJump and WLex. Each feature family is individually removed from the system containing all of them to evaluate its contribution. The increase in AER when removing a feature family reflects its importance. The basic family of features obtains an AER of 13.3% (a relative error reduction of 32.8% over IBM4), which confirms the results reported in (Ayan and Dorr, 2006a).

Adding the new feature families further improves the AER: with these extra features the error rate falls down to 12.8%. While Group1 features families have a positive contribution, their impact on AER varies. Both the AAlign feature family and data partitioning (using WPOS feature family) have high positive contributions, since removing any one of them significantly worsen the AER. Other feature families, including AFert and WLex, have a less important impact, while, in our experiments, WMono does not produce any improvement. While AJump and ADist do not help here, there contribute to improvements when introducing two additional symmetrical alignments, namely "Union" and "Stack", in the "Group1+Union+Stack" configuration. This could be explained by the fact that ADist and AJump are engineered to capture characteristics of a symmetrical alignment. Hence, enhanced performance can only be seen when using "Union" and "Stack" configurations, in which additional symmetrical alignments are used to extract features.

4.9.4.2 Second feature group

The performance of the second group of features (Group2) is compared to a different baseline. The baseline for this experiment is built using the best configuration from the previous experiment without stacking, namely Group1+Union.

This baseline (Group1+Union) uses IBM1 alignments as input, and only $\ell^2 = 0.01$ regularization is applied, without additional ℓ^1 regularization. The threshold is $\rho = 0.5$ meaning that there is no bias towards neither precision nor recall. Discriminative training data is a subset of the IBMA-train containing 2K sentences. The entire NIST 5M parallel sentences are use to train the IBM models. The performance of MaxEnt-baseline is shown in Table 4.7.

The difference with the previous section (4.9.4.1) is the number of training sentences used for MaxEnt training (here, 2K instead of 10K in Section (4.9.4.1).

The following experiments help investigating the effect of the remaining features using the Group1+Union baseline. All features in Group2 help improving both recall and precision. They can be divided in two classes according to their discrimination power; first come WPreff, WSuff, WProb and WFreq with about 0.5 AER reduction each, then come AMultd, WIdent, WPunct and WSim with about to 0.2 AER reduction each. Including all the Group2 features improves AER by 1.6 over the Group1+Union baseline.

Features	Pr%	Rc%	AER%
Basic	90.0	83.7	13.3
Group1	91.6	83.2	**12.8**
− cond/WPOS	89.1	81.6	14.8
− AAlign	88.9	83.4	14.0
− AFert	91.7	82.2	13.3
− WLex	91.7	82.7	13.1
− preserved	91.9	82.9	12.9
− AJump	92.0	82.9	12.8
− ADist	92.1	82.9	12.8
− WMono	92.3	82.8	12.8
+ Union	91.9	83.1	12.7
Group1+Union+Stack	92.4	83.0	**12.6**
−AJump	92.0	82.9	12.7
−ADist	92.0	83.0	12.7

Table 4.6: *Precision, recall and AER results for combining two IBM4 models with different features configurations. basic: features found in literature; Group1 = basic + new features; and Union indicate using the union alignment in input.*

Alignment Model	Pr%	Rc%	AER%
Group1+Union baseline	87.9	69.4	22.4

Table 4.7: *Precision, recall and AER results for the Group1+Union baseline. IBM1 alignments are used as input.*

4.9.5 Precision-Recall Balance

Using the same Group1+Union baseline (Section 4.9.4.2), different thresholds are used to test different balance point between precision and recall. Thresholds between 0.1 and 0.9 shift precision from 81.9 to 92.7 and recall from 72.5 to 64.7.

The lowest AER for the MaxEnt-baseline (22.4) is achieved at $\rho = 0.5$.

4.9.6 Regularization

In this experiment we test the effect of adding ℓ^1 regularization to the small $\ell^2 = 0.01$ used in the Group1+Union baseline (Section 4.9.4.2). Recall that ℓ^1 penalty yields sparse parameter vectors by settings many weights to zero, and hence admits automatic feature selection.

The results in Table 4.8 show that aggressive features pruning with high values of the ℓ^1 regularizer, results in improved precision and recall (hence AER). The biggest AER reduction of 1.2 points (at $\ell^1 = 3$) is attainable while discarding 97% of the features.

4.9.7 Search Space and Window Size

This experiment aims at assessing the effect of increasing the size of the set of input links by considering different window sizes $w = 0, 1, 2$.

Table 4.9 shows that increasing the size of the search space, using larger windows around the current link (e.g. w = 1), reduces the AER by 2.3. When exploring a bigger set of input links, the model is able to retrieve more links, improving the recall by 12.6 points. But it makes more mistakes, since it has to make more decisions, which degrade precision by 9.9

Regularization	Pr	Rc	AER	# active features
baseline: $\ell^2 = 0.01$	87.9	69.4	22.4	501238
+ ℓ^1=0.1	86.7	69.4	22.9	92590
+ ℓ^1=0.5	88.0	69.9	22.1	50380
+ ℓ^1=1	88.8	70.2	21.6	35268
+ ℓ^1=2	89.3	70.3	21.3	19610
+ ℓ^1=3	89.4	70.4	21.2	13806
+ ℓ^1=4	89.3	70.3	21.3	10704
+ ℓ^1=5	89.4	70.0	21.5	8528
+ ℓ^1=6	89.1	70.2	21.5	7334

Table 4.8: *The impact of different values for ℓ^1 regularization.*

Window Size	Pr	Rc	AER
baseline: W=0	87.9	69.4	22.4
W=1	78.0	82.0	20.1
W=2	77.2	81.6	20.6

Table 4.9: *The impact of different sets of input links controlled by the size of the window.*

Input Training Corpus Size	Pr	Rc	AER
baseline: 5M	87.9	69.4	22.4
30K	85.9	64.0	26.7
130K	87.2	66.1	24.8
1030K	87.3	68.3	23.4

Table 4.10: *The impact of different input alignments quality determined by the size of their training data used for generative input alignments. IBM1 alignments are used as input.*

points. The majority of links added by widening the alignment window are inactive, which intuitively worsen the imbalanced data problem. However, its effect is still taken over by the gained boost in recall. It should be mentioned that using sub-sampling methods to address the imbalanced data problem did not deliver better performance.

4.9.8 Input Alignments Quality

To evaluate the model's sensitivity to the quality of input alignments, we exploit the fact that training IBM alignments with less data results in alignments with degraded quality: we train IBM model 1 with MGIZA using corpora of different sizes (30K, 130K, 1030K). Each of these alignments is then used as an input to build a discriminative system. The resulting systems are then compared to the baseline, which is build using IBM model 1 alignment trained on the entire 5M parallel corpus. The baseline's AER drops from 22.4 to 26.7 for the worst input alignment (IBM1 trained on 30K).

Stacking helps correcting errors in the baseline and improves its AER by 1 point by enhancing both recall and precision.

Feature	Weight
$a_{i,j} = \text{active} \wedge \text{WPref}(f_i) = \text{Al\$} \wedge \text{WPref}(e_j) = \text{el}-$	1.7313
$a_{i,j} = \text{active} \wedge \text{WPref}(f_i) = \text{Anh} \wedge \text{WPref}(e_j) = \text{tha}$	1.6652
$a_{i,j} = \text{active} \wedge \text{POS}(f_i) = \text{CC} \wedge \text{POS}(e_j) = \text{CC}$	1.4559
$a_{i,j} = \text{inactive} \wedge \text{WPunc}(f_i, e_j)$	1.2070
$a_{i,j} = \text{active} \wedge \text{MGIZA\_HMM}(f_i, e_j) = \text{active}$	0.7639

Table 4.11: *Sample of selected features with high weights*

4.9.9 Model and Feature Selection

As described in Section 4.4, the use of ℓ^1 regularization yields a sparse model where the most useful features have been selected during the training step. Some of these features are shown in Table 4.11. The first binary feature indicates whether the Arabic word starts with the prefix "*Al*" while the English word begins with the prefix "*el*". This feature indeed embeds a rule of thumb to translate Arabic proper nouns, and is sufficient to ensure correct alignments for all the related occurrences in the test set.

The second feature, shown in Table 4.11, encodes the punctuation mismatch and prevents to align punctuations with regular words. With this feature, the model prefers to leave a punctuation unaligned, rather than aligned with a regular word. This decision is generally the best if a punctuation cannot be aligned with another punctuation.

Even if most of the selected features are related to the input generative models HMM and IBM1 (40% of the features), a more global study shows that all classes are represented in the final model and so are useful for alignment. Moreover, it is worth noticing that 90% of the selected features are conditioned on current POS tags.

4.9.10 A Comparison with Weighted Matrix Based Alignments

In weighted matrix based approaches, which are described in Section 2.3.2, instead of using the Viterbi alignment, a score is computed for each link in the alignment matrix. We also discussed several methods to obtain the final alignment including thresholding which we use. In this experiment we use a subset of the NIST parallel data exploited in the previous section. AER results for four different baselines and the MaxEnt aligners are shown in Table 4.12.

4.9.10.1 Viterbi IBM and HMM models

We include the baseline of the previous section for comparison. For this reason we use Viterbi alignments produced by the multi-threaded alignment toolkit MGIZA++[11] (Gao and Vogel, 2008), symmetrized with GDFA. These models are also used as input to the MaxEnt alignments. The IBM4 line of this system represents the standard baseline.

4.9.10.2 N-best heuristic

Liu et al. (2009)[12] proposes a simple method to obtain link scores which consists of averaging link occurrences over MGIZA++ N-best alignments produced by the IBM model 4. No improvement in AER are observed for this method which is not surprising since the 10-best alignments of IBM4 does not contain enough variation to produce predictions that would be different from the Viterbi alignment. This methods resembles MBR decoding with the difference that only a small number of alignments (10-best) are considered.

[11] http://www.kyloo.net/software/doku.php/mgiza:overview
[12] http://www.nlp.org.cn/~liuyang/wam/wam.html

	Alignment		AER 30K	AER 130K
Generative	**MGIZA++**	HMM	28.35	26.77
		IBM4	24.97	23.30
	10-best	IBM4	24.92	23.26
	PostCAT (Ganchev, Graca, and Taskar, 2008)	Bijective	22.53	20.49
		Symmetric	22.48	20.83
Discriminative	**CRF** (Niehues and Vogel, 2008)	HMM	25.39	23.65
		IBM4	23.51	22.04
		HMM+IBM 1,3,4	21.03	19.65
	MaxEnt	HMM	17.61	16.42
		IBM4	15.61	14.32
		HMM+IBM 1,3,4	14.69	13.92

Table 4.12: *Comparison of five matrix-based word aligners: MGIZA++, 10-best, PostCAT, CRF and MaxEnt, in terms of AER. Two training corpus of different sizes (30K / 130K) are considered.*

4.9.10.3 PostCAT

In Section 2.3.1.2, we have described the Posterior Constrained Alignment Toolkit, which is an open source, freely available[13], implementation of the model described in (Graça, Ganchev, and Taskar, 2007; Ganchev, Graca, and Taskar, 2008; Graça, Ganchev, and Taskar, 2010). This model is an efficient and principled way to inject rich constraints on the posteriors of latent variables into the EM algorithm, allowing it to satisfy additional, otherwise intractable, constraints.

When applying constraints such as a *symmetry* or *bijectivity* on a regular HMM alignment, it delivers models that are comparable in accuracy to the IBM4 model, and under which the statistics needed to estimate posteriors can still be collected efficiently. This allow us to construct weighted matrices with posteriors estimated over constrained HMM models, by calculating for each link, the average of the posterior given by two HMM models in both translation direction, a method referred to as the *soft union* symmetrization.

Both symmetric and bijective constraints help improve performance for the standard HMM. Compared to unconstrained EM, about 20% reduction in AER is achieved over HMM and about 10% over IBM4.

4.9.10.4 CRFs

The alignment matrix is modeled with a CRF, the graphical structure of which is quite complex and contains many loops (Niehues and Vogel, 2008)[14]. Therefore, neither training nor inference can be performed exactly, and the loopy belief propagation algorithm is used to approximate the posteriors.

CRF can incorporate the predictions of other alignments as features, therefore we use the same set of alignments used with MaxEnt. However, unlike MaxEnt, CRF does not use these alignments to prune the set of input links. Therefore, there no constraints on the recall that can be achieved.

[13] http://www.seas.upenn.edu/~strctlrn/CAT/CAT.html
[14] We thank J. Niehues (KIT) for sharing his implementation.

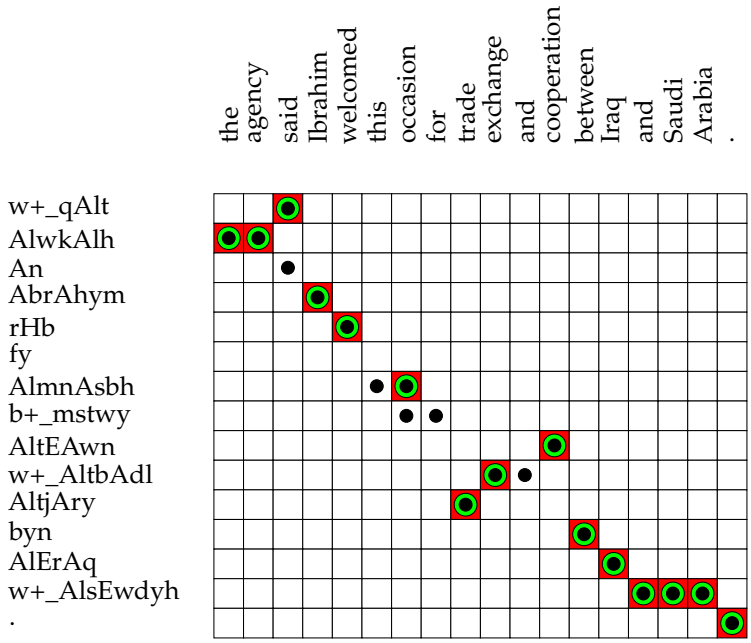


Figure 4.8: *Comparison between manual alignments (big circles), IBM4 alignments (small circles), and MaxEnt alignments (dark rectangles).*

In addition to the model structure, the CRF approach differs from our MaxEnt model in two aspects:

- We train MaxEnt to optimizes the log-likelihood, whereas the CRF which we use is trained to minimizing the AER in addition to the log-likelihood;
- While both models use the same set of features, MaxEnt turns real-valued features into discrete ones using unsupervised equal frequency interval binning.

Combining the two HMM models (similarly IBM4) using a CRF instead of the symmetrization heuristic improves the AER but yields poorer performance than the PostCAT alignments, unless all IBM and HMM models are used by the CRF.

4.9.10.5 MaxEnt

Using HMM models or IBM models or any combination thereof, MaxEnt models produce the best AER, outperforming all the baselines with a large margin.

While these results are not surprising for generative models, they are less understandable for CRF. We suspect that this is due to the fact that the CRF implementation does not benefit from IBM alignments to control the set of links to be predicted; furthermore using real-valued features is sub-optimal as compared with discretization schemes; and finally the structure of the CRF is complicated which leads to approximate training and inference procedures.

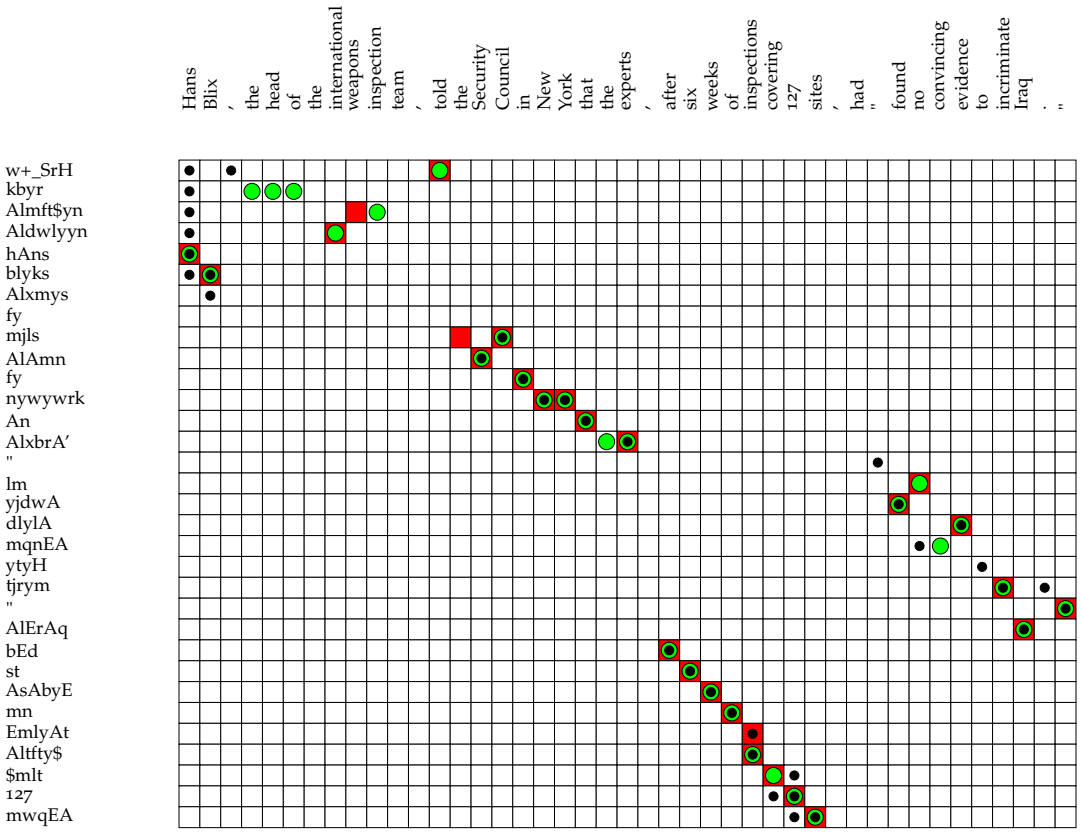


Figure 4.9: *Comparison between manual alignments (big circles), IBM4 alignments (small circles), and MaxEnt alignments (dark rectangles).*

4.10 Error Analysis

Using our MaxEnt model we have drastically reduced the AER, and achieved a good performance on the IBMAC manual corpus. In this experiment, we seek to understand what kind of errors the model is still committing. We looked at several sentences from the IBMAC corpus test set, and compared the alignments obtained by MaxEnt ($\rho = 0.5$) and by IBM4, and manually compared them to the hand annotations. We illustrate some of the types of errors we encountered using few examples. In Figure 4.8, the MaxEnt recovers from IBM4 errors and produce a perfect alignment. Looking at Figure 4.9 of a more difficult sentence pair, we notice that IBM4 is not very precise, and it misses several valid links. MaxEnt improves both the precision and the recall, but it still makes some errors. Some recall errors, such as missing the link "`convincing-mqnEA`", can happen because the link was not present in the input link set, or no sufficient evidence for this link is available, due to data sparsity. Errors can also be related to many-to-many alignments, in which more context is necessary to capture the alignment. This is the case for "`the head of * * * inspection- kbyr Almft$yn`"[15]. In this example, the manual annotation contained links for "`the head of - kbyr` which is missed by MaxEnt. This translation is valid only if it is followed by "`Almft$yn`" and this context is

[15]Note that * is a gap

required to capture the alignment. Some errors are due to the ambiguity in the alignment, where the reference annotation resolves the ambiguity in a different way than the automatic alignment. This is the case for "`the Security Council - mjls AlAmn`" where "`the`" is left unaligned in the reference. The same mismatch is produced for another example "`a fourth team - fryq rAbE`", where the reference did not contain a link for "a", whereas MaxEnt correctly aligns it to "fryq". This ambiguity is inherent in the alignment. This kind of errors is frequent for English preposition which are are always separate words, whereas many of them are implicit in Arabic. This is the case for "`the north of the country - $mAl AlblAd`" where the MaxEnt model drops the alignment of the preposition "`of`". Therefore, many of the alignment errors committed by the MaxEnt model results from the fact that the alignment problem is ill-defined.

4.11 Summary

In this chapter, we have presented a discriminative word alignment model based on the maximum entropy framework. First, we have reformulated the alignment problem as a binary classification task, in which links are predicted individually. Then we have augmented the model with a stacked inference layer in order to better account for the structured nature of the alignment. Stacking indirectly takes the interaction between links into consideration with reduced complexity in comparison to its CRF counterpart. The model parameters are trained to maximize the regularized log-likelihood using elastic-net combination of ℓ^1 and ℓ^2 penalties. Given the model, decoding is performed by thresholding the link posteriors using a hyper-parameter tuned to maximize the AER. The features used by the model cover the the words in the parallel sentence and their PoS tags; and the predictions of other alignment models. In this context, our formulation can be seen as an alignment combination method in which the union of several input alignments is used both to restrict the set of input links and to provide feature functions. Instead of using the a symmetrization heuristic to make local and arbitrary alignment decisions, we use model scores optimized on the entire corpus.

We have provided an extensive empirical study of the intrinsic quality of the alignments produced by our MaxEnt model, when compared to manual alignments. We have obtained state-of-the-art results in terms of the AER, outperforming several alignment models including the generative IBM and HMM models, and discriminative CRF-based model. Since we use the output of generative models to prune the set of input links, we have provided a study of the oracle AER and show that the set of input links we have designed includes almost all the links found in manual alignments. We have shown that the ℓ^1 regularization is very helpful. It performs feature selection and results in sparse and interpretable models while decreasing the AER. We present a careful study of the impact of several novel features and on the performance and show that the prediction of the generative models are important.

The main conclusions of this chapter is that the quality of the alignments can be drastically improved, using simple and inexpensive models, using a small amount of annotated data.

By combining only IBM model 1 and HMM alignments using a MaxEnt model, we obtained a drastical reduction in AER compared to the state of the art IBM model 4. However, these improvements are limited by the capacity of the input alignments to spot the correct links. While simple heuristics can greatly boost this capacity, only slight enhancement in performance is achieved.

In the next chapter, we will evaluate the MaxEnt alignment model in the context of a phrase-based SMT system and show that improvements carry on to the translation quality.

CHAPTER 5

Maximum Entropy Word-Based Alignment Models in Machine Translation

Machine translation is one of the most important application of bitext alignment. State-of-the-art SMT systems, discussed in Chapter 3, are phrase-based, meaning that the translation unit is the phrase The main source of knowledge in such systems is the phrase table, which represents the bilexicon of the translation system. The phrase table gathers the set of the source phrases that are considered by the system, their related target phrases and the scores which evaluate the translation association. Typically, the phrase-based bilexicon is built from generalized phrase alignments, as discussed in Section 2.5.2. MaxEnt alignment models described in the previous chapter were word-based, which means that an additional extraction step is required to obtain the required set of phrase pairs.

In this chapter, we describe two such extraction procedures, compare their performances and their interaction with the different word alignment models. We focus on the MaxEnt model described in Chapter 4. The first method is the standard extraction heuristic and the second is based on weighted alignment matrices. The findings of this chapter where originally published in Tomeh et al. (2010); Tomeh, Allauzen, and Yvon (2011a); Tomeh, Allauzen, and Yvon (2011b); Tomeh et al. (2011a).

5.1 Phrase Table Construction

Building a phrase table from a parallel corpus constitutes the *translation model training* phase, and is usually performed in two main steps (cf. Section 3.2.4):

1. For each training sentence pair, a set of source-target phrase pairs is first extracted.
2. Phrase pairs accumulated over the entire training corpus are collected and evaluated using relative frequencies estimates. Additional scores, based on lexical probabilities are also used. The collection of phrase pairs and their scores constitutes the translation model (aka the phrase table).

The extraction step amounts to computing a generalized phrase alignment (cf. Section 2.5). In Section 2.5.2.1, we have presented the standard extraction heuristic, which extracts the phrase pairs that are consistent with the underlying word alignment. This is the most

Algorithm 1 Phrase Table Construction

Input: Parallel Corpus $\{\tilde{\mathbf{e}}_k, \tilde{\mathbf{f}}_k\}_{k=1}^{\tilde{N}}$
Output: Translation Model $\mathcal{T}$

1: Initialize the phrase table $\mathcal{P} = \{\}$
2: **for all** sentence pairs in the training parallel corpus $(\mathbf{f}, \mathbf{e}) \in \{\tilde{\mathbf{e}}_k, \tilde{\mathbf{f}}_k\}_{k=1}^{\tilde{N}}$ **do**
3: Construct the alignment matrix $\mathbf{A} = align(\mathbf{f}, \mathbf{e})$
4: Construct the set of admissible phrase pairs
$\mathcal{P}_{\mathcal{A}} = \{(\mathbf{p} = i_1...i_2, \mathbf{r} = j_1...j_2) : 1 \leqslant j \leqslant M, 1 \leqslant i \leqslant N, \text{and}(\mathbf{p}, \mathbf{r}) \text{ satisfies the constraints } \mathcal{C}_{\mathcal{A}}\}$
5: $\mathcal{P}_{\mathcal{E}} = \{\langle(\mathbf{p}, \mathbf{r}), c(\mathbf{p}, \mathbf{r}, \mathbf{A}), \eta(\mathbf{p}, \mathbf{r}, \mathbf{A})\rangle : (\mathbf{p}, \mathbf{r}) \in \mathcal{P}_{\mathcal{A}}\}$, where c is an evaluation function, and η is a counting function
6: $\mathcal{P}_{\mathcal{S}} = \{x : x \in \mathcal{P}_{\mathcal{E}}, x \text{ satisfies selection criteria } \mathcal{C}_{\mathcal{S}}\}$
7: $\mathcal{P} = \mathcal{P} \cup \mathcal{P}_{\mathcal{S}}$
8: **end for**
9: **for all** $\langle(\mathbf{p}, \mathbf{r}), c(\mathbf{p}, \mathbf{r}, \mathbf{A})\rangle \in \mathcal{P}$ **do**
10: $\mathcal{T} = \mathcal{T} \cup \{\langle(\mathbf{p}, \mathbf{r}), \phi(\mathbf{r}|\mathbf{p}), \phi(\mathbf{p}|\mathbf{r}), lex(\mathbf{r}|\mathbf{p}), lex(\mathbf{p}|\mathbf{r})\rangle\}$

$$\phi(\mathbf{r}|\mathbf{p}) = \frac{count(\mathbf{r}, \mathbf{p})}{\sum_{\hat{\mathbf{r}}} count(\hat{\mathbf{r}}, \mathbf{p})}, \tag{5.1}$$

where $count(\mathbf{p}, \mathbf{r}) = \sum_{\mathbf{A}} \eta(\mathbf{p}, \mathbf{r}, \mathbf{A})$, and

$$lex(\mathbf{r}|\mathbf{p}, A_{\mathbf{r},\mathbf{p}}) = \prod_{i=1}^{length(\mathbf{r})} \frac{1}{|\{j : (i,j) \in A_{\mathbf{r},\mathbf{p}}\}|} \sum_{\forall (i,j) \in A_{\mathbf{r},\mathbf{p}}} w(e_i|f_j), \tag{5.2}$$

11: **end for**

commonly used approach in practice. We have also presented the weighted-matrix based approach, which takes the alignment distribution into accounts. We now present a general algorithm to build the phrase table, that unifies these two phrase pairs extraction methods.

5.1.1 A General Framework

Algorithm 1 sketches a general approach to construct the translation model $\mathcal{T}$, by extracting and scoring phrase pairs from a parallel corpus $\{\tilde{\mathbf{e}}_k, \tilde{\mathbf{f}}_k\}_{k=1}^{\tilde{N}}$.

For all sentence pairs $(\mathbf{f}, \mathbf{e})$ made up of M source words and N target words, we would like to enumerate all possible phrase pairs $(\mathbf{p}, \mathbf{r})$ and assign each of them a score (c) that quantify their quality and can be used as a phrase pairs selection criterion (for instance, by applying a threshold on this score).

Yet, extracting *all* possible phrase pairs found in the training corpus would cause practical problems, as i) the related growth in the number of extracted phrase pairs could dramatically slow down decoding; ii) the number of target phrases extracted for each source phrase would increase, and the simple scoring method based on relative frequencies is not capable of distinguishing between them. Therefore, a selection procedure is required.

The final step is to score the selected phrase pairs, and to store them in the phrase table. These scores usually include a translation probability ϕ, estimated using relative frequencies over the training corpus, where each occurrence of a phrase pair is evaluated using the counting function η. They also include lexical weights lex, based on lexical translation probabilities w, as a smoothing method to improve the estimates computed for rare phrase

pairs. A valuable, and relatively easy to acquire, source of information is the word alignment represented by the alignment matrix **A**, which is consulted at different steps of this algorithm: during filtering, evaluation and scoring of phrase pairs.

5.1.2 Viterbi-Based (Standard) Approach

The most common instantiation of this framework (Koehn, Och, and Marcu, 2003) considers a binary alignment matrix **A**, where each cell represents a binary variable indicating whether the associated words are aligned or not. The matrix is usually obtained by applying the symmetrization heuristic to two Viterbi alignments, one for each translation direction. The alignment constraints $\mathcal{C}_{\mathcal{A}}$ are defined so that extracted phrase pairs $(\mathbf{p}, \mathbf{r})$ are consistent with **A**:

$$\forall (i, j) \in \mathbf{A} : (j \in [j_1, j_2] \wedge i \in [i_1, i_2]) \vee (j \notin [j_1, j_2] \wedge i \notin [i_1, i_2]).$$

Consistency means that words inside a phrase pair can not be aligned to words outside it. The selection criteria $\mathcal{C}_S$, used in line 6 of the algorithm, may be grammatical such as retaining only the phrases that correspond to tree constituents or to chunks. In most SMT systems, phrases are limited to a certain maximum length, which improves the efficiency. All selected phrase pairs are evaluated and counted using $c = \eta = 1$. The standard instantiation is illustrated in Figure 5.1.

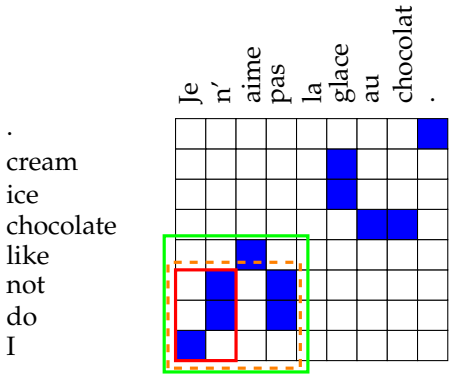


Figure 5.1: *Viterbi-based standard instantiation of the extraction heuristic. Only the solid-line rectangle represents a consistent phrase pair with the underlying word alignment.*

5.1.3 WAM-based Instantiation

The training of the translation model is thus decomposed as a modular pipeline, the components of which can be developed independently. The resulting modularity comes at the price of possible error propagation between consecutive steps: errors in the Viterbi word alignment can propagate to the phrase pair extraction and the probability estimation steps.

Since the standard instantiation ignores alignment probabilities, it tends to be sensitive to alignment errors. An erroneous link, as unlikely as it may be, can prevent the extraction of many plausible phrase pairs. Furthermore, the extracted phrase pairs are all considered of equal quality, regardless of how much evidence the alignment matrix provides for them.

This problem can be alleviated by feeding more information from word alignment into the pipeline. For this purpose, a structure called the *Weighted Alignment Matrix* (WAM) (Liu et al., 2009), which compactly encodes the distribution of all possible alignments of a sentence pair,

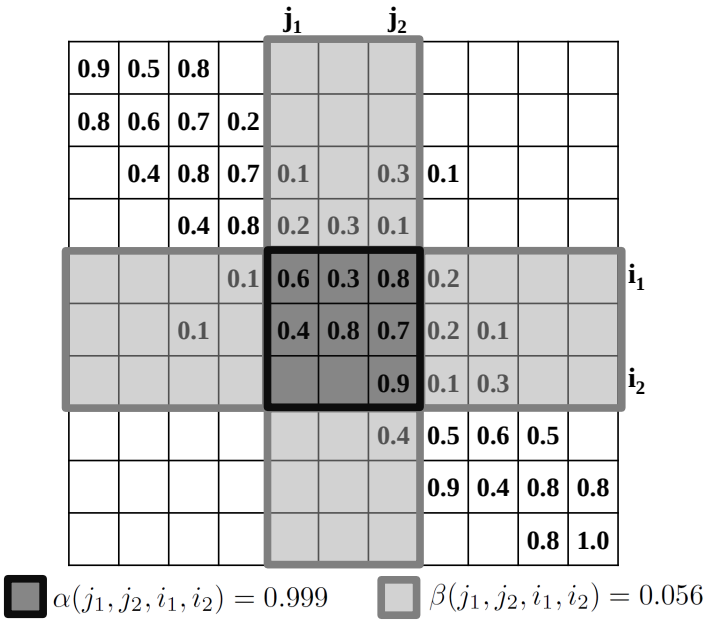


Figure 5.2: *Computation of fractional counts:* $\eta(\mathbf{p}, \mathbf{r}) = \alpha(i_1, i_2, j_1, j_2) \times \beta(i_1, i_2, j_1, j_2)$. *Empty cells have zero probability.*

can be used to extract and to score phrase pairs. Each cell in this matrix corresponds to a pair of (source, target) words; the associated real value measures the quality of the alignment link. Therefore, a weighted matrix encodes, in linear space, the probabilities of exponentially many alignments. In the weighted matrix $\mathbf{A_w} = \{p((i,j)|\mathbf{f},\mathbf{e}) : 1 \leqslant i \leqslant N, 1 \leqslant j \leqslant M\}$, each possible link is weighted by a score $p((i,j)|\mathbf{f},\mathbf{e})$ quantifying the confidence assigned to it by the alignment model.

5.1.3.1 Evaluation and counting functions

The use of a weighted matrix enables to design more informative evaluation and counting functions, which can help mitigate the error propagation problem. To incorporate alignment posterior probabilities when computing fractional counts for a phrase pair, all possible alignments should be enumerated. Unlike for N-best (Venugopal et al., 2008) or HMM (Gispert, Pino, and Byrne, 2010) alignments, this is unrealistic for a weighted matrix. Instead, we follow (Liu et al., 2009) and use link probabilities to compute a fractional count, interpreted as the probability that the phrase pair satisfies consistency constraints.

Given a weighted alignment matrix $\mathbf{A_w}$ and a phrase pair $(\mathbf{p}, \mathbf{r})$, two regions (in gray on Figure 5.2) are identified: $in(j_1, j_2, i_1, i_2)$ and $out(j_1, j_2, i_1, i_2)$ which respectively represent links *inside* and *outside* (on the same rows and columns) of a phrase pair. We use MaxEnt (cf. Chapter 4) to compute the posterior probabilities $p((i,j)|\mathbf{f},\mathbf{e})$, from which the probability that two words are unaligned is obtained as $\bar{p}((i,j)|\mathbf{f},\mathbf{e}) = 1 - p((i,j)|\mathbf{f},\mathbf{e})$. We can now compute, for the inside region, the probability that there is at least one word inside one phrase aligned to a word inside the other phrase as:

$$\alpha(i_1, i_2, j_1, j_2) = 1 - \prod_{(i,j) \in in(i_1, i_2, j_1, j_2)} \bar{p}((i,j)|\mathbf{f},\mathbf{e}). \tag{5.3}$$

Similarly for the outside region, we compute the probability that no word inside one phrase

is aligned to a word outside the other phrase:

$$\beta(i_1, i_2, j_1, j_2) = \prod_{(j,i) \in out(i_1, i_2, j_1, j_2)} \bar{p}((i,j)|\mathbf{f}, \mathbf{e}). \tag{5.4}$$

Finally, the same function is used for evaluation and counting ($c = \eta$) and defined as the product of these two probabilities:

$$\eta(\mathbf{p}, \mathbf{r}) = \alpha(i_1, i_2, j_1, j_2) \times \beta(i_1, i_2, j_1, j_2). \tag{5.5}$$

5.1.3.2 Alignment constraints and selection criteria

Weighted alignment matrices admit flexible alignment constraints and selection criteria. Thresholding enables to better tune the balance between the number of extracted phrase pairs and the accuracy of their assigned scores. A possible choice for $\mathcal{C}_A$, adopted in our experiments, is to require that at least one link inside the phrase pair has a probability $p((i,j)|\mathbf{f}, \mathbf{e}) > t_a$. Similar constraints could be applied on links outside the phrase pair. Likewise, $\mathcal{C}_S$ admits only phrase pairs with an evaluation score greater than a threshold $c(\mathbf{p}, \mathbf{r}) > t_p$, subject also to the phrase length limit.

5.1.3.3 Translation model scores

While the phrase translation probability estimated as ϕ (see step (10) of Algorithm 1) can be applied unchanged when using the fractional counts η, the lexical scores lex have to be modified to incorporate link probabilities. The main difference is the computation of the lexical probabilities $w(e_i|f_j)$ and $w(f_j|e_i)$, which are usually computed using relative occurrence frequencies (Koehn, Och, and Marcu, 2003). Instead of simply counting every occurrence once $[count(e_i, f_j) = 1]$, link probabilities provided by the weighted matrix are used as fractional counts: $count(e_i, f_j) = p((i,j)|\mathbf{f}, \mathbf{e})$ (Liu et al., 2009). Using fractional counts for c, η and w yields a more accurate evaluation of phrase pairs depending on the context of the sentence-pair in which they occur, hence a better estimation of their scores.

In Section 2.5.2.1 we discussed other possible ways to compute the evaluation function c. In the next chapter we present a novel technique for this purpose.

5.2 Experiments

In this section, we evaluate the MaxEnt alignments model by measuring their impact on translation systems performance using the two phrase pairs extraction procedures detailed above. Phrase-based translation systems are built using Moses[1] (Koehn et al., 2007) with SRILM[2] for language modeling. The target side of the parallel data is used to train a 4-gram back-off language model. MERT (Och, 2003) is carried on to tune the parameters of the translation system on the NIST MT'06 test set. The "standard" set of features (Section 3.2.3) is used across all the experiments.

Translations are evaluated on the NIST MT'08 test set. As in Chapter 4, data made available by the NIST'09 constrained evaluation track is used to train the generative alignment models, while the IBM Arabic-English aligned corpus (IBMAC) (Ittycheriah, Al-Onaizan, and Roukos, 2006) provides us manual word alignments. IBMAC includes a training set that we split into disjoint train and dev sets, used respectively for training and tuning the discriminative models. We perform our experiments in four translation tasks of different size. The large scale experiments in Section 5.2.1.1 use all the data. Three smaller tasks of are also considered

[1] http://www.statmt.org/moses/
[2] http://www-speech.sri.com/projects/srilm/

in Sections 5.2.1.2 and 5.2.2. These tasks use subsets of the data containing respectively: 30K, 130K, 1030K parallel sentences. In all our experiments, Arabic data are pre-processed using MADA+TOKAN as described in Section 4.8.2.

The experiments in this chapter are organized in two main sections. We start by comparing the translation performance of different alignment models including our MaxEnt aligner, when the standard Viterbi-based extraction heuristic is used. In the second Section (5.2.2), we replace the extraction heuristic with the method based on weighted matrices. We compare its performance with the standard heuristic and we also compare the MaxEnt model with other methods in the weighted matrix context.

5.2.1 Viterbi-Based Extraction

Using the standard extraction heuristic, we first compare MaxEnt alignments with the standard generative IBM and HMM models in a large scale experiment settings, and we study the correlation between the alignment quality and the translation performance. Then, using the three sub-tasks described earlier, we compare the MaxEnt model to additional baselines including a wider range of generative and discriminative models. We also study the relation between different alignment characteristics (Guzman, Gao, and Vogel, 2009) and the translation performance.

5.2.1.1 Large scale systems

Experiments are carried out using a large-scale Arabic to English phrase-based system developed for the NIST MT Eval'09 in the constrained training condition[3].

MaxEnt vs. IBM and HMM models Although large-scale phrase-based systems tend to be robust to word alignment errors (Lopez and Resnik, 2006), improvements in translation quality are still attainable.

As explained, our approach can be used to learn the symmetrization heuristic from the data. Table 5.1 shows that the combination of two IBM4 models by a Maximum Entropy model results in an absolute gain of 0.6% BLEU point over a combination of these two alignments by a heuristic. Another interesting result is that a discriminative alignment considering the computationally inexpensive IBM1 and HMM alignments as an input, performs, at least, as well as the standard IBM4-GDFA[4].

Table 5.1[5][6] shows BLEU, AER and $F_{0.3}$ scores obtained for GIZA++ GDFA alignments and various discriminative alignments. In terms of BLEU, the best performing discriminative alignment is the one combining eight generative models (IBM1, HMM, IBM3 and IBM4 in both directions) with a threshold $\rho = 0.4$: it achieves a BLEU score of 41.1% and a 0.7% absolute improvement over the best generative model.

Results of discriminative alignments suggest that they systematically improve translations over generative models. However, the impact of the features set and of stacking is unclear: using the features configuration that gives the best AER (denoted *best* 5.1) leads to slight improvements in BLEU (0.1%) over the *basic* feature set proposed by (Ayan and Dorr, 2006a). Stacking does not seem to improve translation performance, even though it slightly improves the AER (Table 4.3). This suggests that in order to have significant improvements in BLEU, relatively large improvements in AER should be achieved.

[3] http://www.itl.nist.gov/iad/mig/tests/mt/2009/

[4] Runtime needed to train the MaxEnt model is negligible and labeling is linear in the size of the corpus, which is in total faster than training IBM3 and IBM4 models (minutes vs. hours/days)

[5] In this table we show BLEU scores for thresholds that give either the best AER (usually $\rho = 0.7$) or the best $F_{0.3}$ (usually $\rho = 0.4$)

[6] Only Group1 features are used to build alignments for this experiments.

				AER	$F_{0.3}$	BLEU
GDFA Alignment						
IBM1				29.4	70.8	39.3
HMM				22.6	78.9	40.0
IBM3				20.7	81.0	40.5
IBM4				19.8	82.6	40.4
Discriminative Alignment						
model	*features*	ρ	*stack?*			
IBM4 [2]	basic	0.6	✗	13.3	85.2	40.8
		0.3	✗	15.2	85.7	40.9
		0.4	✗	14.2	86.0	40.9
	best	0.7	✗	12.7	85.6	40.7
		0.4	✗	13.5	86.6	41.0
		0.7	✓	12.6	85.6	40.8
		0.4	✓	13.3	86.6	40.7
IBM1+HMM [4]	best	0.4	✗	14.4	85.1	40.5
ALL [8]	best	0.4	✗	12.9	87.4	**41.1**
		0.7	✗	12.1	86.3	40.7
		0.4	✓	13.0	87.5	40.9

Table 5.1: *Translation results in BLEU for different GIZA++ and discriminative word alignments. best corresponds to Group1+Union and st. to stacking. In bold is the best system's score.*

It is also worth noting that for a given input alignment set, BLEU results are not very sensitive to differences in threshold values. For example, when combining two IBM4 alignments, using the basic features configuration, threshold values 0.3, 0.4 and 0.6 produce comparable BLEU results of 40.9, 40.9 and 40.8, respectively. This suggests that picking-up an acceptable value for the threshold does not require an exhaustive search for the optimal solution. It is also interesting to note that, for generative GDFA alignments, moving from IBM1 to IBM4 improves performance by 1.1% BLEU point. This improvement is quite small compared to the increase of time required to train the model and much smaller than what has been observed for other language pairs.

Correlation between AER and BLEU The relationship between alignment quality and machine translation performance has been discussed in (Langlais, Simard, and Veronis, 1998; Fraser and Marcu, 2007b). Vilar, Popovic, and Ney (2006) show that translation quality can be improved while decreasing AER. Giving less weight to alignment points that connect multiple aligned words improves correlation (Davis, Xie, and Small, 2007). Training word alignment models with translation-related loss functions, although computationally expensive, also improves the correlation (Lambert, Banchs, and Crego, 2007).

In the experiments discussed in (Fraser, 2007), the AER exhibits a low correlation with translation quality as measured by BLEU, due to the distinction between possible and sure links. When this distinction is suppressed, correlation is improved as it is the case for balanced F-Measure (Fraser and Marcu, 2007b). The F-Measure achieves a better correlation by means of additional flexibility by varying the trade-off between precision and recall.

In order to gain more insights on these issues, we conducted a systematic experiment, whose results are depicted in Figure 5.3, to evaluate the correlation between the BLEU score

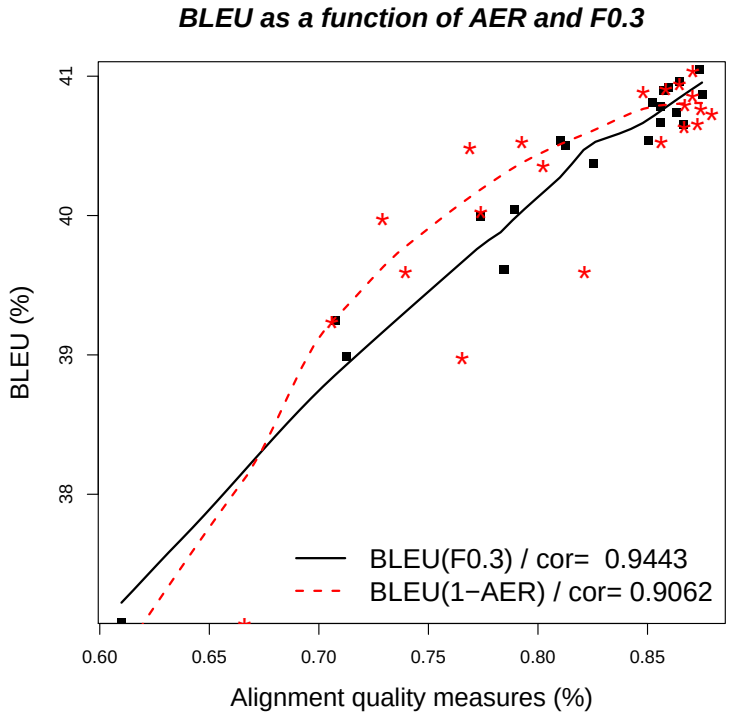


Figure 5.3: *Correlation between BLEU and alignment quality measures.*

and various alignment metrics. We have built 22 systems using different word alignment methods or parameters (11 systems were using a generative alignment method with different symmetrization heuristic; 11 systems were using a discriminative alignment method with different feature sets or input alignments or thresholds). We then computed F_α for all values of $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$ (with 1−AER corresponding to $\alpha = 0.5$).

The highest correlation between the alignment metric and the translation metric is obtained for $F_{0.3}$: their correlation, measured by the Pearson coefficient, is 94.34%. Note that the AER metric that is usually used to assess the performances of alignment methods also correlates well with BLEU but at a lower coefficient of 90.62%.

The high correlation suggests that the alignment quality remains good predictor of the translation quality. However, it is not the only factor as we will discuss in the nest section. The reason is that alignments go through an extraction step before being used in by the translation system, and many characteristics of the resulting phrase table are not accounted for by the AER, such as the size of the phrase table and its coverage. These characteristics are related to alignment sparsity, the number of unaligned words, the number of gaps in phrase pairs, etc. The threshold ρ is used to control the density of the resulting alignments and therefore shifting the balance between precision and recall. Alignments with lower ρ are denser, and hence tend to have higher recall. Translation results show that alignments with an higher recall tend to perform better, suggesting that recall is preferable over precision and is the best predictor of the translation quality. Consequently, BLEU is expected to correlate better with measures favoring recall like $F_{0.3}$.

5.2.1.2 A study of alignment characteristics

In this experiment, we aim to study further the relation between additional components of the MaxEnt aligners and their impact not only on AER but on translation quality. Given the way we use alignment to train translation systems, two sources of errors may affect phrase pairs consistency. On the one hand, word alignments are error prone, and they sometimes fail to detect word-level translations which carry on to the extracted phrase pairs. On the other hand, the extraction heuristic achieves generalization by combining aligned words into phrases, and growing over unaligned ones around them. This can be helpful to treat cases where no word-level alignment exists, such as in the translation of propositions, idiomatic expressions and compound words. However, since this heuristic operates locally on a sentence level and only make heuristic decisions, it could easily extract noisy phrases, especially when given a wide marge of freedom by leaving plenty of words unaligned.

Since word alignments are the only constraints on the extraction heuristic, they become the only way to control both sources of errors mentioned earlier: by setting on a good balance between the alignment quality and the number of unaligned words. Therefore, regardless of the resulting AER, the tradeoff between precision and recall for word alignments has a great impact on the quality of the extracted phrases.

For instance, let us consider the case of a perfect precision (all links are correct), but with a low recall (not all word-level correspondences are detected). Then the alignment matrix is sparser than it should be, and the proportion of unaligned words results in many phrase pairs, with moderate scores (since they allow for multiple translations which over-flatten the probability distribution). The human-perceived quality of resulting phrases also degrades (Guzman, Gao, and Vogel, 2009). In the other case, with a high recall and low precision, the alignment matrix is denser than it should be, and generalization fails, with fewer and over-deterministic phrase pairs. Thus, the quality of a phrase table depends on the interaction between the quality of word alignments (precision and recall) and the sparsity of the alignment matrix: the number of unaligned source or target words, and the resulting gaps. In this experiments, we aim to understand how the interaction between many alignment characteristics determines their performance in translation.

The results in Table 5.2 must be viewed in light of this discussion. IBM1 is an example of alignments where performance is apparently affected by both types of errors are apparently affecting its performance. Compared to a manual alignment: IBM1 (1) produces poor alignment quality with low precision and recall, which causes the extraction of erroneous phrase pairs, and (2) leaves too many words unaligned, which adds to the noise in extracted phrases. More complex generative models are more efficient: HMM and IBM4 improve both precision and recall, while aligning more words. These enhancements lead to the extraction of less noisy phrase pairs, which eventually perform better. IBM4 for example, improves BLEU by 2 points over IBM1 for the smallest task, and 1.4 points for the biggest one.

ME-Group1-Features alignments have different profile than IBM models. They have much better recall and precision (75% for IBM4 to 92.7% for the stacked baseline). Thus, the quality of extracted phrase pairs should be improved significantly since they are based on better word-level correspondences, and the first source of errors is limited. But on the other hand, these alignments tend to be more sparse that than manual alignments (9% less links) and than IBM4 alignments (21% less links), which causes more extraction errors, degrading the quality of phrase pairs. Otherwise stated, the improvement in phrase table quality due to AER improvement, is almost canceled out by increasing the number of gaps. This explains why discriminative models achieve overall small improvements over IBM models.

ME-All-Features systems are similar to the previous ones but use better feature engineering and ℓ^1 regularization. They achieve comparable alignment quality (precision, recall), but they are able to align more words, and to decrease the percentage of gaps. This results in higher BLEU scores on all the three tasks.

Enlarging the search space (using a window of size 1) allows for a significant increase in

Alignment Characteristics						Phrase-pairs Characteristics					BLEU		
Recall	*Precision*	*AER*	*#links*	*#UnalignSrc*	*#UnalignTgt*	*#Phrases*	*avgSrcGap*	*%GaplessSrc*	*avgTgtGap*	*%GaplessTgt*	*30K*	*130K*	*1030K*
Manual alignments													
100.0	100.0	0.0	16171	2655	3457	86642	0.68	56.5	0.83	47.6	-	-	-
Generative baselines (GDFA): *IBM1, HMM, IBM4*													
70.2	71.0	29.4	16394	3032	4752	72369	0.98	47.6	1.28	36.9	36.0	39.2	40.6
73.7	81.4	22.6	17985	1967	3524	74782	0.63	62.8	0.96	46.6	37.5	40.5	41.5
75.1	86.1	**19.8**	18715	1422	2460	60029	0.34	75.4	0.57	61.0	38.0	41.1	**42.0**
ME-Group1-Features IBM1-HMM ($\rho = 0.6$), +Stacking ($\rho = 0.5$)													
90.7	82.0	13.9	14733	3435	4851	119303	1.04	44.2	1.29	33.9	38.0	41.3	42.3
92.7	81.5	**13.2**	14953	3371	4542	122412	0.99	44.8	1.13	34.4	38.2	41.4	**42.3**
ME-All-Features *IBM1-HMM ($\rho = 0.5$), +Window ($\rho = 0.8$), +Stacking ($\rho = 0.7$)*													
91.4	82.7	13.2	15215	3197	4552	106490	0.93	47.5	1.18	36.8	38.2	41.4	42.8
89.7	86.6	11.8	17143	3436	4019	107063	1.05	43.7	1.14	39.2	37.4	40.5	41.8
93.1	86.5	**11.2**	16054	3008	4173	108825	0.91	46.6	1.15	38.9	38.5	41.7	**42.9**

Table 5.2: *Characteristics of alignments in terms of their quality compared to gold standard, number of links and unaligned source/target words. The number of extracted phrases is included with the average number of gaps per source/target word, and the percentage of gapless phrases. These statistics are computed using the IBMAC test set 4.1. Finally the quality of alignments in terms of their impact in BLEU for three different MT tasks. Th is the threshold.*

recall (from 82.7% to 86.6%) with slightly degraded precision, which improves the AER. These alignments change the balance between unaligned source and target words, with respect to the previous systems: more *source* words, and *less* target words are aligned, yielding a comparable phrase table size. This configuration is harmful and results in about 1 BLEU point loss on all tasks. An interesting result is presented in the next line, when adding a stacking layer to the system with the enlarged search space. Stacking fixes the problem with precision, without harming recall, improves the over all quality of the alignment, and reduces the number of unaligned source words, shifting the balance back in the right direction. This system achieves the lowest alignment error rate of 11.2%, and the best BLEU score on all three tasks, with significant improvements over the generative baselines (for the biggest task).

5.2.2 Weighted Matrix Based Extraction

In this set of experiments, we have two goals:

- compare the standard Viterbi based extraction and scoring method to the method based on weighted alignment matrices; and
- contrast different approaches to fill the weighted matrices: our MaxEnt method and different baselines including MGIZA++, PostCAT and CRF alignments.

We use the same training data as described in Section 5.2.1.2, namely a subset of the LDC resources made available for the NIST MT'09 constrained track. In order to validate the obtained results on training corpora of varying sizes, we consider two training conditions, one with 30K parallel sentence pairs, and another with 130K. For each condition, we recall AER from Section 1.7.1.1 and provide the BLEU scores on the test set, along with the size of the obtained phrase tables.

			Translation task:	30K					130K				
			Translation model construction:		Standard(i)		WAM(ii)			Standard(i)		WAM(ii)	
	Alignment			AER	BLEU	PT	BLEU	PT	AER	BLEU	PT	BLEU	PT
Generative	MGIZA++	HMM		28.35	35.01	3,6	-	-	26.77	39.15	9,7	-	-
		IBM4		24.97	35.90	2,4	-	-	23.30	40.18	6,5	-	-
	10-best	IBM4		24.92	35.78	2,4	36.21	3,0	23.26	40.00	6,6	40.43	8,5
	PostCAT	Bijective		22.53	36.62	3,3	36.94	10,2	20.49	40.08	9,1	40.61	29,5
		Symmetric		22.48	36.69	2,9	36.96	10,7	20.83	40.24	8,5	40.43	30,2
Discriminative	CRF	HMM		25.39	35.93	4,6	36.50	11,9	23.65	39.56	12,6	40.00	31,2
		IBM4		23.51	36.07	3,4	36.93	8,4	22.04	40.34	8,7	40.32	21,3
		HMM+IBM 1,3,4		21.03	36.34	3,7	37.10	8,4	19.65	40.14	9,8	40.35	21,3
	MaxEnt	HMM		17.61	36.90	6,7	37.48	11,7	16.42	40.47	17,7	40.84	30,0
		IBM4		15.61	37.17	5,5	37.52	9,6	14.32	41.04	14,5	41.13	25,0
		HMM+IBM 1,3,4		14.69	37.12	5,2	37.92	8,6	13.92	40.82	13,4	41.08	22,2

Table 5.3: *Comparison of five word aligners: MGIZA++, 10-best, PostCAT, CRF and MaxEnt, in terms of AER, BLEU scores and Phrase Table size in millions (PT). We compare the standard to the WAM-based instantiation of Algorithm 1. Two training corpus of different sizes (30K / 100K) are considered.*

In the word alignment step, we experiment two configurations of the alignment matrix: (i) a standard alignment matrix, which contains the links of the 1-best alignment; and (ii) the weighted alignment matrix, which is populated with link probabilities. Note that we can obtain a matrix in configuration (i) by thresholding the probabilities in the weighted matrix according to a threshold t_a[7].

Hence, for each word aligner that produces a weighted matrix, we derive two systems: *standard* and *WAM-based* [8]. The two remaining steps depend on the form of the alignment matrix computed in the first step:

- For standard matrices (i) we use the standard heuristic for extraction, and relative frequencies for scoring (Koehn, Och, and Marcu, 2003).
- For weighted matrices (ii), a phrase posterior can be computed and used as a fractional count $\eta(\mathbf{p}, \mathbf{r})$. Only phrase pairs with a fractional count above certain threshold t_p[9] are extracted. The same fractional counts are used for scoring with relative fractional frequencies. In both configurations, only phrase pairs that do not exceed a length limit of 7, on the source or the target side, are retained and scored.

5.2.2.1 Results and discussion

We compare our MaxEnt alignments to the same baselines used in the previous chapter, namely: MGIZA++ (for Viterbi baselines and n-best WAM and discriminative features); PostCAT; and CRF. Table 5.3 displays the results in terms of AER and BLEU.

MGIZA++ In this setting, MGIZA++ refers to deterministic alignment matrices in configuration (i). MGIZA++ IBM4 represents the performance of the standard baseline: one IBM4 alignment in each direction, which are symmetrized with GDFA heuristic. This system delivers competitive BLEU scores of 35.9 and 40.2 on the 30K and 130K respectively, with a much smaller phrase table than all the other systems.

[7] In our experiments t_a is set to 0.5.

[8] Our experiments show that post-processing the weighted matrix to nullify all link probabilities, that are inferior to a threshold t_a, improves the performance. We use $t_a = 0.5$.

[9] In our experiments t_p is set to 0.1.

N-best WAM This setting (cf. Section 4.9.10.2) slightly improves performance over the baseline. Gains of 0.3 BLEU point on the small task and 0.2 on the larger one are achieved. Improvements are only obtained in the weighted matrix configuration, while standard alignments obtained by thresholding the (10-best based) weighted matrix seem to hurt performance for the selected threshold (0.5). Phrase tables obtained using these systems are only slightly larger than the baseline, which might explain the small improvement. The N-best system achieves comparable AER to the MGIZA++ baseline.

PostCAT Our experiments use Geppetto[10] (Ling et al., 2010), an implementation of the weighted alignment matrix integrated with PostCAT. For the small task, both bijective and symmetric PostCAT alignments, in the standard configuration, outperform MGIZA++ and N-best WAM by $\approx$ 0.8 BLEU point. The weighted matrix configuration performs even better than the standard one and increases BLEU scores by another $\approx$ 0.3 BLEU point. Improvements are persistent but less apparent on the larger task. We notice that the phrase table extracted from the weighted matrix is considerably larger than the standard one (by a factor of at least 3). PostCAT also slightly decreases the AER as compared to the MGIZA++ baseline.

CRF On the small task, the CRF approach achieves improvement up to $\approx$ 0.4 over the MGIZA++ baseline and up to $\approx$ 1.2 over the WAM-based baseline. Using several input alignments as local features seems beneficial: approximately 0.5 BLEU point, for both configurations, is gained when using IBM3 and IBM4 features. Similar tendencies are observed for the larger task, albeit with smaller gains. The performance of CRF is comparable to that of PostCAT, but its translation models are however somewhat smaller. The CRF model is trained in two optimization stages, the first one maximizes the log-likelihood, while the second minimizes the AER starting from the parameters obtained in the first step. Nevertheless, the CRF achieves only modest improvements in AER over MGIZA++ and PostCAT as explained in Section 4.9.10.4.

MaxEnt Discriminative weighted matrices significantly outperform all the previous baselines in both configurations. For the 30K task and for the standard configuration (i): when using only MGIZA++ HMM alignments as input to MaxEnt, we get 1 BLEU point improvement over the standard MGIZA++ IBM4 baseline, and 0.2 point over PostCAT. The extracted phrase table is twice as large as the ones used by the MGIZA++, 10-best or PostCAT. Further improvements are obtained when using IBM4 as input or combining several input alignments, 1.3 BLEU point over MGIZA++ and 0.5 point over PostCAT. MaxEnt based matrices, in configuration (ii), achieve up to 2 BLEU point improvement over MGIZA++ IBM4 and up to 1 point over the best weighted matrix baseline (PostCAT). It is noticeable that this latter improvement is obtained with a smaller phrase table ($\approx$ 25% smaller).

These gains persist for the larger task: MaxEnt in standard (i) configuration is 0.8 BLEU point better than MGIZA++ IBM4, and 0.6 better than PostCAT. In the weighted matrix configuration (ii), these improvements allow us to outperform MGIZA++/IBM4 by nearly 1 BLEU point, 10-best by and approximately 0.7 point, and PostCAT by 0.5 point. As for the size of the phrase table, MaxEnt delivers smaller phrase tables (22,2M) than PostCAT (30,2M), but much larger ones than MGIZA++ IBM4 (6,5M). Unlike all the other systems, MaxEnt drastically decreases the AER, and achieves approximately a 40% relative reduction over MGIZA++ on both 30K and 130K tasks.

5.2.2.2 Discussion

We discuss a bit further the results obtained for the 30K translation task. Similar discussion can be carried out for the 130K task. Figure 5.5 allows us to focus on the Viterbi based

[10] http://code.google.com/p/geppetto/

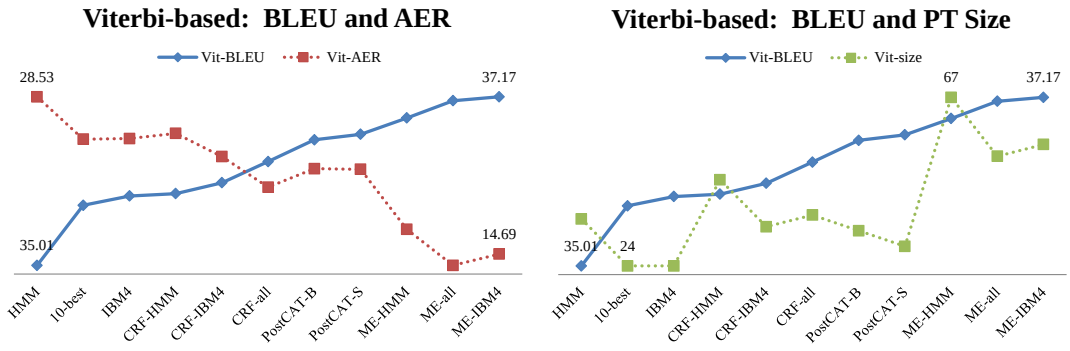


Figure 5.4: *Viterbi-based extraction. BLEU, AER and phrase table size results.*

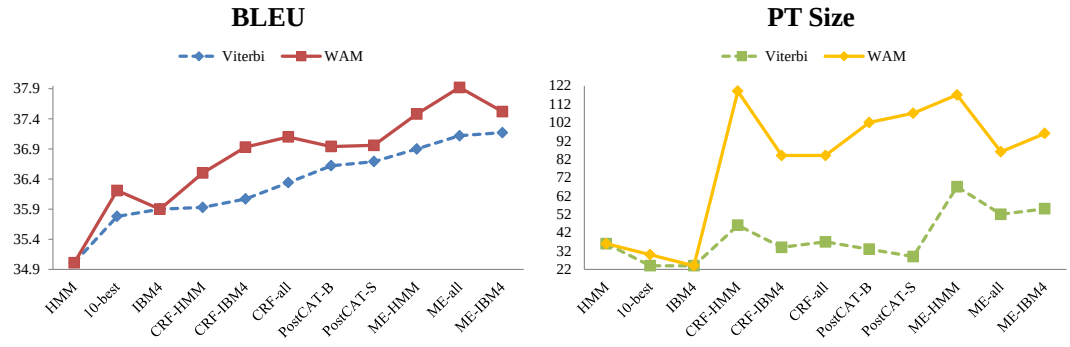


Figure 5.5: *Comparison of Viterbi- and WAM-based extraction in terms of BLEU scores and the corresponding phrase table sizes.*

extraction results for a wide range of alignment models. The plot on the left[11] show that all the MaxEnt models obtain the best results in terms of BLEU and AER scores, which are correlated. The plot on the right shows that the phrase table size and BLEU scores are less directly correlated. This is due to the properties of the HMM model: HMM produce a lower quality alignment as measured by AER, while allowing to extract larger phrase tables.

On the other hand, Figure 5.5 allows us to compare Viterbi- and WAM-based extraction in terms of their BLEU performance and their corresponding phrase table sizes, across different alignment models[12].

The plot on the left show that WAM-based extraction outperforms its Viterbi counterpart for all systems. MaxEnt alignments are preferable in BLEU scores with both methods. It is worth noting that this performance is obtained by producing the largest phrase tables under Viterbi extraction. However, under WAM-based extraction they produce relatively smaller phrase tables comparable in size to other models, while still achieving the best BLEU scores. CRF alignments, while being slightly behind PostCAT with Viterbi-based extraction, catch up when WAM-based is used. This is obtained while producing smaller phrase tables than PostCAT.

[11] Note that AER curves can not be produced for the weighted alignment matrix without thresholding.
[12] Note that HMM and IBM4 do not admit WAM-based extraction.

5.3 Summary

Using a discriminative MaxEnt alignment models helps estimate better translation models: we showed that these improved alignments result in an increase of 0.7 BLEU points for a large-scale Arabic-to-English system. We have also showed that it is possible to achieve, by combining IBM1 and HMM alignments through discriminative training, models that outperform the conventional setting (IBM4 symmetrized alignments), at a much lower computational expense. We finally showed, in a series of systematic experiments, that there is a correlation between the quality of the word alignment measured by the $F_{0.3}$ metric and the BLEU score.

We have also contrasted alignments obtained by the symmetrization heuristic with those obtained by the discriminative matrix model, in the light of their AER and their impact on translation quality as measured by BLEU on NIST MT08 large-scale task. We have analyzed the BLEU results in light of several alignment characteristics and have noticed that finding a better balance between the alignment quality, as measured by its precision and recall, its sparsity, and its number of unaligned words and extracted phrases is necessary to deliver better translation models.

We have finally presented a generic algorithm to construct the translation model from a parallel corpus, for which we described two instantiations: standard and WAM-based. Within this framework we have compared several generative and discriminative word aligners in both instantiations, and showed that the WAM-based outperforms the standard procedure due to its improved use of the word alignment probability distribution as compared to the Viterbi alignments. Using WAM-based extraction, our MaxEnt modeling of the matrix led to approximately 2 BLEU points improvement over the standard MGIZA++ baseline, using a small training corpus and 1 BLEU point using a larger one.

We conclude that improving the quality of word alignment, as measured by its precision and recall, leads to improvements in translation quality. However, other factors are also important, such as the interaction between the sparsity of the alignment and the phrase pair extraction methods.

CHAPTER 6

Supervised Phrase-Based Alignment with Single Class Classification

So far, we considered alignment with word constraint where alignable units are words. Motivated by the fact that single words are not always the best units to capture translation relations (cf. Section 1.2), we relax here this constraint and allow phrases to be aligned directly.

Many problems, such as word lexical ambiguity, fertility and reordering, that had to be addressed explicitly in word translation, are partially solved implicitly by considering longer units. Additionally, generalized phrase-based alignments are required to train PBSMT systems of the type, discussed in Section 3.2.2.

In order to obtain phrase pairs suitable to translation, a special attention should be paid to the conditions under which PBSMT systems produce an appropriate translation:

- Appropriate translations must exist in the search space of the decoder; and
- The model score must be (positively) correlated with translation quality.

The first condition mainly depends of the coverage of the phrase translation candidates that are stored in the phrase table. A maximal coverage can be achieved by including all possible phrase pairs encountered in the training corpus: in this setting, the model scores are the only information used to select suitable translations during decoding.

However, given the sheer number of possible phrase pairs, the vast majority of which are in fact irrelevant, taking all possible phrase pairs into account is impractical, and most practical methods for constructing phrase tables start with a phrase alignment and extraction step where the quality of each phrase pair is estimated, and where phrase pairs that look erroneous are filtered out.

Section 2.5.2.1 reviewed several approaches for this purpose, based on the weighted phrase-based matrix illustrated in Figure 2.12. The standard methodology (Koehn, Och, and Marcu, 2003) fills the matrix with binary scores deduced from the underlying word alignment and discards all phrase pairs that are not consistent with it. This technique, however, does not let the user control the size of the resulting phrase table.

More flexibility is gained by using pruning techniques that need to be applied *a posteriori* as in (Johnson et al., 2007; Tomeh, Cancedda, and Dymetman, 2009), where a second scoring

step is used to filter large phrase tables. An alternative is to assign each phrase a smooth score in the interval $[0, 1]$ and then use thresholding as discussed in Section 5.1.

These alignments are error-prone and they are obtained as the result of complex optimization programs maximizing an objective function (the likelihood) that correlates only indirectly with translation quality. The same applies to the computation of feature functions used in the translation model score during decoding. However the combined model is enhanced during *tuning* to better correlate with translation quality, where feature weights are set so as to optimize an automatic quality measure, such as BLEU.

As an attempt to improve these procedures, we study in this chapter a new method to compute the scores used for phrase extraction. The presented phrase extraction procedure exhibits several desirable properties:

- can straightforwardly handle arbitrary feature functions;
- have a direct relationship to translation quality; and
- give the user a finer control over the size of the phrase table.

This study has both practical and methodological implications. From a practical perspective, the scenario we consider is the use of a small set of parallel sentences, from which we would like to extract as much phrases as possible, so as to ensure the largest possible coverage. In this setting, finding better ways to score phrases might prove necessary. From a more methodological perspective, our goal is to better understand the properties that make a good phrase pair. To fulfill these goals, we reformulate the extraction problem in a supervised learning framework.

Extracting or discarding a phrase pair is indeed a binary decision, which can be learned, using a set of labeled training examples. We would like to make such decisions based on the expected utility of phrase pairs in a translation pipeline. This leaves us with two issues: (a) defining an operational notion of utility, and (b) finding examples of useful and useless phrase pairs with respect to this definition.

As discussed below, a good approximation of (a) will be to consider that phrase pairs participating in derivations of good translations are useful; such phrase pairs can be collected by looking at good derivations of test data, and will provide us with sets of *positive* training examples. Obtaining negative examples proves more challenging, and would require to examine all the (non-optimal) derivations of our test data, which is clearly unrealistic. A nice workaround is to use *single-class classification* (Tax, 2001) techniques, which aim at learning concepts in the absence of counter examples, by distinguishing one class of (positive) instances from all other possible instances.

Such techniques can handle arbitrary feature functions to represent candidate phrase pairs, thus making the extraction procedure more robust to alignment errors. A useful by-product of the model is also the computation of an *accuracy-based feature*, analogous to the proposal in (Penkale et al., 2010).

The rest of this Chapter is organized as follows. In Section 6.1, we motivate the formulation of phrase pair extraction as a single-class classification problem and describe a practical extraction pipeline. The *One-Class SVM (OC-SVM)* (Schölkopf et al., 2001) and the *Mapping Convergence (MC)* (Yu, 2005) algorithms, which are used to train the single-class classifier are presented in Section 6.2. In Section 6.3, we describe the oracle decoder used to label positive examples. Our feature functions, describing various facets of a phrase pair are detailed in Section 6.4. Experiments are reported in Section 6.5 before we conclude in Section 6.6 with related work and a summary of the chapter. The findings of this chapter were originally published in (Tomeh et al., 2011b).

6.1 Supervised Phrase-Pair Extraction

As previously mentioned, we would like to score phrase pairs in a way that is directly related to the translation quality, and can take advantage of several feature functions, that account for different aspects of phrase pairs quality and cast as a back-off for the alignment models. A straightforward solution is to shift the extraction procedure to the supervised learning paradigm.

6.1.1 Single-Class Classification (SCC)

In this section, we would like to learn the binary decision of extracting or discarding a phrase pair as a supervised classification problem, in which we aim to discriminate *useful (positive)* from *useless (negative)* phrase pairs in a translation perspective. The model score can also be used as a new feature function to score candidate phrase pairs.

An obvious way to recast this problem as a supervised classification problem requires labeled training examples of both classes, which implies an understanding of what makes a useful phrase pair.

Such a task is tricky even for humans. From a phrase-based model point of view, a phrase pair is useful if (1) each phrase is an appropriate translation of the other and (2) it combines well with neighbor phrase pairs to produce a good translation. This means that the evaluation of the quality of phrase pairs is dependent on the context in which it is used. While scores associated to a phrase pair provide evidence regarding the validity of the translational equivalence, the combination aspect is more difficult to judge without involving the translation process itself. We therefore define a positive phrase pair as one that participates in best scoring derivations of good translations, which are easy to obtain: it is sufficient to find one context in which a phrase pair is useful to be able to label it as positive. For this purpose we constrain the PBSMT decoder to produce the reference translation (cf. Section 6.3). Unfortunately, negative phrase pairs can not be identified the same way because this requires to examine all the possible translations where they occur and make sure none is acceptable: we should verify that the phrase pair is not useful in any context. This is obviously impractical.

A particularly appropriate solution in this setting is the Single-Class Classification (SCC) approach, which seeks to distinguish one class, for which positive instances exist, from data in a universal set containing one or several other classes, for which no sample is available.

We assume that the very large set of all possible phrase pairs contains a small set of positive examples $P = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$ completed with a large unlabeled set $U = (\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u})$. The ratio between positive and negative phrase pairs is *unknown*, but assumed to be unbalanced where positive is the minority class. The data is thus generated from multiple distributions where positive phrase pairs are assumed to be drawn from a certain distribution, whereas negative phrase pairs are drawn from any combination of the other distributions.

6.1.2 Phrase Translation Model Training Algorithm

The algorithm described in this section takes as input a parallel corpus, and uses an oracle decoder and some other resources to compute phrase pair features; it outputs a phrase translation model, in the form of phrase table.

In *step (1)*, we build the set U of phrase pairs that are going to be considered by the algorithm. For each one of them, a set of feature functions is calculated. U can be constructed naively by considering all possible phrase pairs found in the parallel corpus, or by applying some prior knowledge, such as word alignments, to filter this set;

In *step (2)*, the set (or a subset) of phrase pairs in U is used to build a phrase table, using the calculated features as scores. An oracle decoder (Section 6.3) uses this phrase table on a held-out parallel corpus, to produce the best phrasal derivations of this corpus. The best

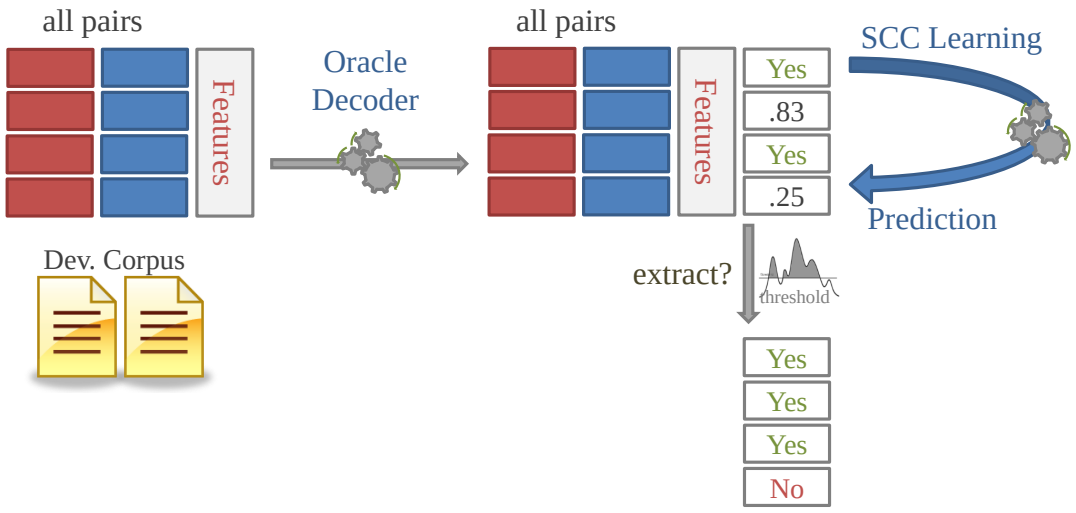


Figure 6.1: *Phrase pairs scoring and extraction using single-class classification.*

derivation is the one that maximizes a combination of model score and translation quality metric. All phrase pairs involved in these derivations are labeled as positive phrase pairs in P;

In *step (3)*, we seek to generalize beyond the scope of the phrase pairs that were actually used by the decoder. As discussed in Section 6.1.1, the oracle decoder acquires a subset of positive phrase pairs, that we want to expand, by learning its characteristics using a classifier. In the next section, we introduce a One-Class Support Vector Machines (OC-SVM) algorithm, designed to learn from positive examples P only, by estimating the support of their distribution (Schölkopf et al., 2001). In practice, this approach is sensitive to the choice of features and parameter settings and is likely to underfit or overfit easily (Raskutti and Kowalczyk, 2004). Therefore, we use the Mapping Convergence (MC) algorithm (Yu, 2005), a semi-supervised framework, which, in addition to learning from positive examples, exploits unlabeled data to improve the accuracy of the classifier.

In *step (4)*, the best classifier trained in step (3) is applied to the entire unlabeled set $(U - P)$, estimating to what extent they resemble the positive samples, and which ones ought to be extracted. The distance to the decision boundary (the hyperplane in the SVM feature space) is interpreted as a confidence measure, and used for two purposes: it is thresholded to extract phrase pairs; and also stored into the final phrase table as an accuracy-based feature function, similar to (Galron et al., 2009; Penkale et al., 2010). The final phrase table contains all the phrase pairs labeled as positive either by the oracle decoder or by the learned classifier. Any subset of the features, in addition to the standard phrase translation probabilities (normalized frequencies) can be used to score phrase pairs in the output phrase table.

This algorithm is sketched in Figure 6.1.

6.1.3 Balancing Precision and Recall

Training a phrase translation model requires to address precision and recall issues, following an information retrieval scheme (Deng, Xu, and Gao, 2008).

High precision requires that the extracted phrase pairs should be accurate, while high recall seeks to increase coverage by extracting as many valid phrase pairs as possible.

On the other hand, there are valid translation pairs in the training corpus that are not learned due to word alignment errors (Tomeh, Allauzen, and Yvon, 2011a). The algorithm presented here attempts to circumvent alignment errors and increase accuracy by integrating multiple features and combining them discriminatevly.

At the same time, the threshold on the classifier score enable us to easily balance between precision and recall, and introduces an additional parameter that can be tuned via *grid search*, so as to get optimal performance for each specific translation task.

6.2 Learning the Single-Class Classifier

SCC seeks to distinguish one class of data from universal set of multiple classes, for example distinguishing personal homepages from other web pages. It is assumed that a reasonable sample of the negative data is hard to acquire, and learning should be performed from only positive data. Several approaches addressed the problem, of which we consider OC-SVM which generalizes the ν-SVM to the unsupervised case; and an iterative algorithm called MC that improves over OC-SVM by exploiting unlabeled data.

6.2.1 One-Class SVM (OC-SVM)

Extensions of SVM have been proposed to allow learning in single-class setting, such as the One-Class SVM (OC-SVM) (Schölkopf et al., 2001), and the Support Vector Data Description algorithm (SVDD) (Tax and Duin, 1999). which are shown to be equivalent when data vectors are scaled to unit length (Tax, 2001). We use OC-SVM, in which the optimization problem is formulated as in ν-SVM parameterization (Schölkopf et al., 2000). In the binary case, ν lets one effectively control the number of support vectors and eliminate the need for the regularization constant C from the original SVM formulation. Similarly in the one-class case, ν allows us to control the fraction of outliers.

Learning from positive data only can be seen as a density estimation task. Knowing the density enables to solve any problem that is based on this data. However, density estimation is a difficult task, therefore, OC-SVM aims to solve an easier problem: given some dataset drawn from an underlying probability distribution, it estimates a subset $\mathcal{S}$ of the input space, such that the probability of a test point drawn from outside of $\mathcal{S}$ equals some *a priori* specified value between 0 and 1. This is done by estimating a function f which is positive on $\mathcal{S}$ and negative on its complementary. Informally put, the computed function f is supposed to capture regions in input space where the probability density lives (its support), that is, most of the data (positive phrase pairs) will live in the region where f is nonzero (Schölkopf et al., 2001).

The functional form of f is given by a kernel expansion; it is regularized by controlling the length of the weight vector in an associated feature space. The expansion coefficients are found by solving a quadratic programming problem. This is done by performing a sequential optimization over pairs of input patterns. This algorithm is an extension of the support vector algorithm to the case of unlabeled data (Schölkopf et al., 2001).

The main idea behind OC-SVMs is to create a hyperplane in the feature space where the projections of data points are separated from the origin with a large margin. This is illustrated in Figure 6.2. The data is separable from the origin if there exists a vector $\mathbf{w}$ such that $\forall i, K(\mathbf{w}, \mathbf{x}_i) > 0$, where K is a kernel function. In such a case, there exists a unique supporting hyperplane. This is always true for the special case of a Gaussian (Radial Basis Function) kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|}$, where the properties $\forall i, j, K(\mathbf{x}_i, \mathbf{x}_j) > 0$, and $\forall i, K(\mathbf{x}_i, \mathbf{x}_i) = 1$ result in all mappings being in the positive orthant.

As pointed out in (Schölkopf et al., 2001), there exists a strong connection between OC-SVMs and binary classification. Assuming we have a parameterization $(\mathbf{w}, \rho)$ for the supporting hyperplane of a data set $\{\mathbf{x}_1, \ldots, \mathbf{x}_l\}$, where ρ is the margin, then $(\mathbf{w}, 0)$ is the parameterization of the maximally separating hyperplane for the labeled data set:

$$\{(\mathbf{x}_1, +1), \ldots, (\mathbf{x}_l, +1), (-\mathbf{x}_1, -1), \ldots, (-\mathbf{x}_l, -1)\}$$

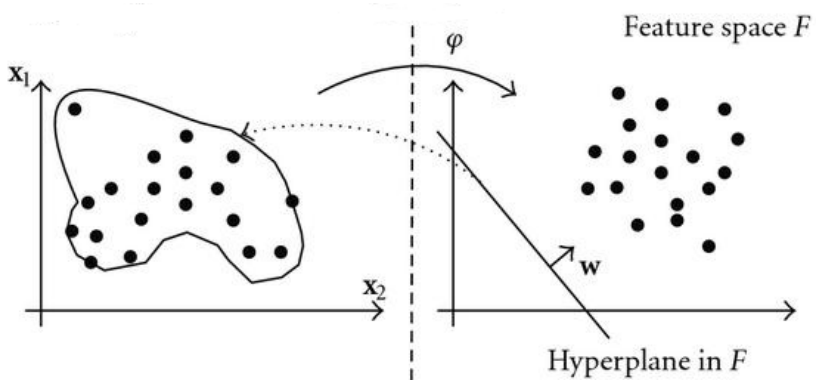


Figure 6.2: *One-class SVM: non-linear mapping of training examples from the input space to the feature space.*

. Also, assuming that we have a maximally separating hyperplane parameterized by $(\mathbf{w}, 0)$ for a data set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}, (y_i \in \{\pm 1\})$ and with a margin $\rho/\|\mathbf{w}\|$, we know that the supporting hyperplane for the unlabeled dataset $\{y_1\mathbf{x}_1, \ldots, y_l\mathbf{x}_l\}$ is parameterized by $(\mathbf{w}, \rho)$.

For the non-separable case, margin errors in the binary setting correspond to outliers in the one-class case. This connection allows us to reuse the optimization problem of ν-SVM to find the supporting hyperplane, such that in the one-class setting, ν represents an upper bound on the fraction of outliers (margin errors) and a lower bound on the fraction of support vectors.

6.2.2 Mapping Convergence (MC)

OC-SVM draws a nonlinear boundary around the positive data set in the feature space using two parameters: ν (to control the number of outliers) and γ (to control the smoothness of the boundary). They have the same advantages as regular SVMs, such as efficient handling of high dimensional spaces and nonlinear classification using the kernel trick.

The problem with OC-SVM is its tendency to draw a very conservative tight boundary fitting the positive data. To illustrate the problem, we consider Figure 6.3, adopted from (Yu, 2005; Hovelynck and Chidlovskii, 2010), which contains a data set U composed of seven data clusters in 1-D, of which only the dark middle one is positive. Everything is unlabeled except for the dark subset of positives. The optimal boundary is represented by the dashed lines. OC-SVM would end up fitting the positive data on (b_p, b'_p). This could be the result of overfitting the data due to its inability of using any knowledge about the distribution of U. Intuitively, the desirable boundary should cover P and should separate it from the remaining data. Such boundary is represented by the two vertical dashed lines in Figure 6.3.

Several attempts to take advantage of large sets of unlabeled data have been studied (see (Zhang and Zuo, 2008) for a survey). The *Mapping Convergence (MC)* algorithm (Yu, 2005) assumes the presence of a "gap" between positive and negative points in the feature space and uses it by incrementally marking as negative unlabeled samples using the *margin maximization* property of SVM. MC has been shown to generate as accurate boundaries as standard SVM with fully labeled data. A key intuition of MC is that negative examples can be sorted by their distance to the decision boundary, the farthest ones being called the *strong negatives*.

MC, described in Algorithm 2, is thus composed of two stages: the mapping stage and the convergence stage.

1. In the mapping stage, the algorithm uses OC-SVM (C_1) to compute an initial approximation of strong negatives in U, the set of unlabeled examples.

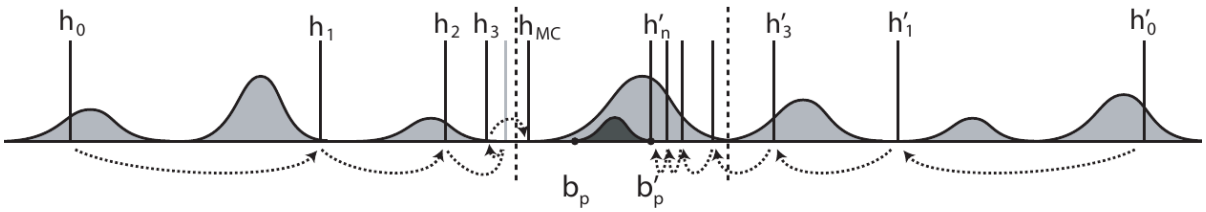


Figure 6.3: *Mapping Convergence in 1-dimensional space*

Algorithm 2 Mapping Convergence (MC)

Input: positive data set P;
unlabeled data set U;
negative data set $N = \emptyset$;
OC-SVM: C_1;
SVM: C_2
Output: boundary function h_i

1: $h_0 \leftarrow$ train C1 on P
2: $\hat{N}_0 \leftarrow$ strong negatives ($\leqslant$10%) from U for h_0
   $\hat{P}_0 \leftarrow U - \hat{N}_0$
3: $i \leftarrow 0$
4: **while** $\hat{N}_i \neq \phi$ **do**
5: $\quad N \leftarrow N \cup \hat{N}_i$
6: $\quad h_{i+1} \leftarrow$ train C_2 on P and N
7: $\quad \hat{N}_{i+1} \leftarrow$ negatives from $\hat{P}_i$ for h_{i+1}
8: $\quad \hat{P}_{i+1} \leftarrow$ positives from $\hat{P}_i$ for h_{i+1}
9: $\quad i \leftarrow i + 1$
10: **end while**

2. Based on this initial approximation, the convergence stage runs iteratively using a binary SVM (C_2) to maximize the margin in order to make a progressively better approximation of negative data.

When no new negatives are discovered, MC converges and the boundary comes to a hold. MC starts by identifying strong negative examples, by using a OC-SVM with a high threshold to favor higher recall and install the initial hypothesis (h_o, h'_o) far from the positive data. Subsequent convergence steps improve boundary (h_i, h'_i) toward the optimal one by adding unlabeled data recognized as negatives to N, then using binary SVM and taking advantage of its margin maximization property, which avoids stopping in an arbitrary gap in the feature space. At the last iteration, the set $\hat{P}_i$ contains the fraction of U that is labeled as positive by the converged MC classifier. The phrase table is then built from phrase pairs un $\hat{P}_i \cup P$

6.2.3 $\hat{P}P$ Measure and Classifier Selection

Cases when not too many positive examples are available, or when too much unlabeled items act as noise would result in incapacity of detecting the gap between positive and negative data in the feature space, which slows down convergence and makes MC over-iterate and overfit.

An example is given in (Hovelynck and Chidlovskii, 2010), where a measure called $\hat{P}P$ is introduced and shown to be effective in detecting convergence and is hence used as stopping criterion.

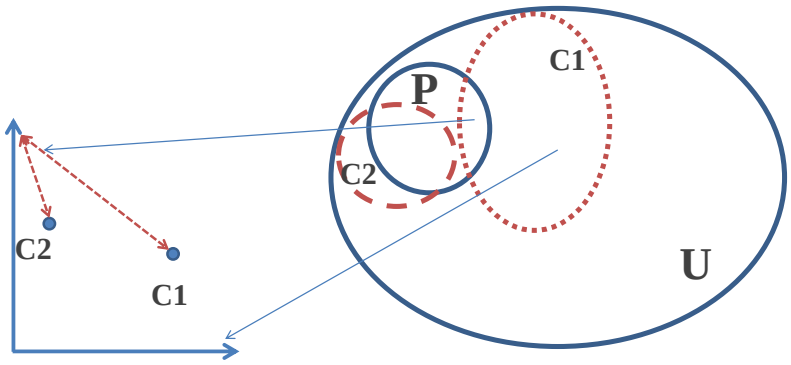


Figure 6.4: *Computing the $\hat{P}P$ measure. The percentage of truly positive points (labeled as positive by a classifier (C1 or C2), and existing in P), are plotted on the vertical axis as a function of the percentage of points that the classifier labeled as positive. We assume that a better classifier should retain a high percentage of P (the truly positive points), while keeping the number of points labeled as positive at minimum. The closest the classifier gets to the upper left corner the better is.*

Here, we follow this approach with a slight modification to incorporate a parameter to control the SVM decision threshold. This parameter regulates the size of the rejected unlabeled data and hence the size of the resulting phrase table.

Hereafter, a classification hypothesis h and a threshold α identify a classifier h_α, using the following decision rule

$$f_{h_\alpha}(\mathbf{x}) = sgn(\delta_h(\mathbf{x}) - \alpha) \tag{6.1}$$

where $\delta_h(\mathbf{x})$[1] is the SVM decision value, on which α acts as a threshold and allows to shift the decision boundary in the feature space. The standard SVM decision function f_h is obtained at $\alpha = 0$.

Every such classifier h_α corresponds to a point in the $\hat{P}P$ plot[2]. On the x-axis, we plot the percentage of the entire data set positively classified $|\hat{P}_\alpha|/|U|$, while on the y-axis, we plot the percentage of positive data positively classified.

In Figure 6.6, each dotted curve depicts the performance of different threshold values α for a certain classifier h_i, resulting from MC iterations. The standard curve (solid line) traces the performance of the standard classifier ($\alpha = 0$) for each MC iteration.

It can be interpreted in a ROC-like fashion. The upper left corner represents a perfect classifier, while points on the diagonal are hypothesis performing random selection from U. The first point in the convergence sequence will be close to the upper right corner: the mapping stage is about selecting a small part of the data set containing only near-certain negatives. Subsequent convergence steps will try to produce a smaller selection, moving leftwards on the curve, while maintaining performance on the positive subset.

Contrary to a genuine ROC curve, the $\hat{P}P$-curve is not continuous, and the step-like behavior of points (x_i, y_i) is not guaranteed, which makes it impossible to calculate the area under the curve (AUC). An alternative is to identify the point in the curve that discards most of the data in U, while keeping a large part of the positive data P. The point on the curve that is closest to (0, 100) is considered as the best classifier. Therefore, a "good" classifier should maximize the recall on the set P while minimizing the number of points labeled as positive.

To assign more importance to precision or recall, the distance measure can be weighted and/or different values of α can be used.

[1] $\delta_h(\mathbf{x}) = \mathbf{w}.\Phi(\mathbf{x}) - \rho$, where $\mathbf{w}$ is the classifier weight vector.
[2] This is shown in Figure 6.6 for the classifier we obtain in our experiments

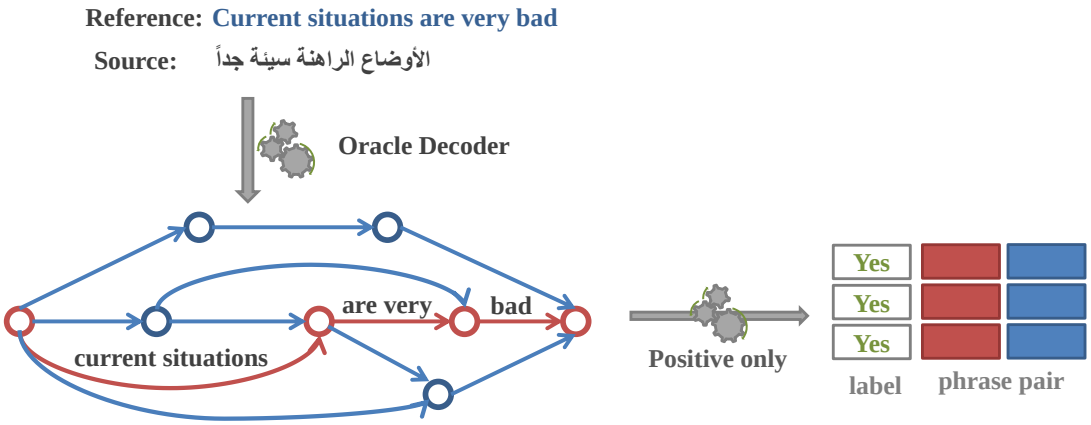


Figure 6.5: *Labeling phrase pairs with an oracle decoder. All phrase pairs that appear in the oracle translation are considered to be positive.*

In every iteration of MC the resulting classifier scores are thresholded with several values of α and the corresponding $\hat{P}P$ points are calculated. Convergence is achieved when no more improvements of $\hat{P}P$ are observed.

MC is similar in spirit to the EM algorithm, which is applied by (Nigam et al., 2000) for text classification tasks. (Nigam et al., 2000) use a combination of EM and a naive Bayes classifier to learning from labeled and unlabeled documents. Zhang and Zuo (2008) surveys the existing method of learning from positive and unlabeled examples.

6.3 Oracle Decoder for Building the Set of Positive Examples

The approach for supervised learning of phrase extraction introduced in Section 6.1.2 relies on a set of positive phrase pairs.

In the PBSMT paradigm, good phrase pairs are required to fulfill two criteria: (1) participate in derivations of good translations with the highest BLEU scores (or another translation quality measure) with respect to some reference translation(s); and (2) have a good intrinsic quality as measured by the phrase-based model score.

This implies that, for identifying positive examples, we need to search among all possible translations, represented as a scored lattice, for the one that jointly optimizes the model score and the translation quality. Once the optimal path in the lattice is found we harvest all the phrase pairs used in the derivation to be labeled as positive and added to P.

There are several methods to find the best path, of which we use two in our experiments. The first method is *constrained decoding*, as implemented in Moses[3]: the lattice is searched for the path with the highest model score that exactly matches the reference, and thus has a local BLEU score of one.

However, if the reference is not reachable, the sentence is discarded. The second method relaxes this constraint by using an oracle decoder that searches for the hypothesis that explicitly optimizes an approximation of the BLEU score at the sentence level as an objective.

We implemented the lattice oracle decoder from (Dreyer, Hall, and Khudanpur, 2007), which, while being less conservative than constrained decoding (all source sentences are decoded), is agnostic about the model score which, therefore, needs to be optimized indirectly by pruning the lattice input of the decoder.

The labeling process is illustrated in Figure 6.5.

[3] http://www.statmt.org/moses/

6.4 Feature Functions

One of the main motivation of this work is to incorporate features into phrase pairs extraction, so as to smooth the conventional, alignment-based, phrase scores. We consider features from the literature (Venugopal, Vogel, and Waibel, 2003; Johnson et al., 2007; Deng, Xu, and Gao, 2008; Tomeh, Allauzen, and Yvon, 2011a; Turchi and Ehrmann, 2011), which evaluate various aspects of the association between a source and a target chunk.

Most features are data-driven and language-independent, based on statistical word alignment and language models. A small set of language-dependent morpho-syntactic features is also used.

6.4.1 Weighted Alignment Matrix (WAM)

WAM feature is a score computed using discriminative Weighted Alignment Matrices presented in the previous chapter 5, and similar to the model-based phrase pair posterior metric described in (Deng, Xu, and Gao, 2008).

Each cell in a weighted matrix (Liu et al., 2009) contains the posterior probability of aligning the corresponding source and target words, as computed by the MaxEnt word aligner from Chapter 4. A phrase pair splits the underlying weighted matrix in two areas: *inside* and *outside* the phrase pair, where *consistent* and *inconsistent* links respectively live. The WAM feature is a score that combines two factors characterizing these areas and quantifies the consistency of the given phrase pair with respect to the entire probability distribution over all possible alignments.

6.4.2 Word Alignments (WA)

These features, similar to (Venugopal, Vogel, and Waibel, 2003; Deng, Xu, and Gao, 2008), evaluate the association between source and target phrases according to the number of consistent and inconsistent word links.

Given a standard alignment matrix obtained by thresholding the weighted matrix, and a phrase pair, a *consistent* link associates words inside the phrase pair boundary, while an *inconsistent* link crosses the phrase pair boundary. This feature is the ratio between the number of consistent links and the sum of the number of consistent and inconsistent links.

6.4.3 Bilingual and Monolingual Information (BI, MI)

BI and MI features are proposed in (Deng, Xu, and Gao, 2008) as measures of the reliability of a phrase pair.

Extracting candidate translations for every phrase to maximize coverage, is not always feasible and might hurt precision. Some phrases could not be accurately aligned due to data sparsity and limitations of alignment models; while other phrases carry no linguistic meaning, such as phrases that are parts of non-compositional phrases or metaphorical expressions. The BI feature addresses the first issue by estimating how reliably the model aligns a phrase pair.

Given a weighted alignment matrix, we calculate the WAM score for all phrase pairs, and normalize them to estimate for every source phrase ($\mathbf{p}$) a conditional distribution over all target phrases: $P_{WAM}(.|\mathbf{p})$. The same computation is performed for every target phrase ($\mathbf{r}$).

The BI score of a source or a target phrase is defined as the entropy of the corresponding distribution. Low entropy implies a high confidence that the source/target phrase can be reliably aligned by the model. Conversely, high value of the entropy signals the impossibility to correctly identify the right alignment.

The MI feature addresses the second issue by evaluating to what extent a certain n-gram is a "good" phrase, by measuring how the choice of the phrase boundaries affects the quality of the phrase. The boundaries of a good phrase are assumed to be the right places to segment

the source sentence. This feature evaluates the quality of the phrase pair boundaries using monolingual language models.

Given a sentence of length N and a history of n words before a boundary (between words i and $i+1$), the forward language model probability $p(*|w_{i-(n-1)} \dots w_i)$ defines a conditional distribution. A similar distribution is defined for the "history" after the boundary, and, in this case, a backward language model is used $p(*|w_{i+(n-1)} \dots w_i)$. The predictive uncertainty (PU) of the boundary between word i and $i+1$ is computed as the sum of the entropy of the forward and backward language models conditional distributions. The larger the predictive uncertainty is, the more likely is the boundary to be located in a "reasonable" place in the sentence.

A good phrase pair is hence characterized with high PU values on its four boundaries, the product of which is the value of the MI feature. This feature captures how well a phrase pair combines with its neighbors to form new parallel sentences.

6.4.4 Statistical Significance (Pval)

Pval feature, as proposed in (Johnson et al., 2007), captures the fact that not all phrase pairs are equally supported by the training data. By including corpus level statistics, this feature gives an overall view of the statistical properties of phrase pair.

For a given phrase pair and a parallel corpus, we compute the source, target and joint source/target frequencies and draw a 2x2 contingency table representing the unconditional relationship between source and target phrases. We then calculate the one-tail p-value of the Fisher's Exact test, interpreted as the probability that the observed table or a more extreme one could occur by chance assuming a model of independence.

We take $|\log(\text{p-value})|$ as the value of this feature: that means that the higher it is, the more significant is the phrase pair.

6.4.5 Morpho-Syntactic Similarity (MS)

MS feature, unlike the previous ones, is language dependent and takes morpho-syntactic information into account.

This feature resembles the measure proposed in (Turchi and Ehrmann, 2011), which captures morpho-syntactic Part-Of-Speech (POS) similarity between source and target phrases.

We use here co-occurrence statistics of source/target POS tags, linked in the word aligned parallel corpus, to build a matching table similar to an IBM model's translation table, which provides association scores between source and target POS tags. The sum of these scores, for aligned words inside the phrase pair, normalized by the number of consistent links, is used as the value of the POS similarity feature. The higher this value is, the stronger the syntactic similarity of the given phrase pair.

6.4.6 Lexical Probability (LEX)

LEX features, as described in (Koehn, Och, and Marcu, 2003), and found in standard phrase tables, use word translation probabilities to quantify the extent to which words inside the phrase pair translate each others. These features are similar to POS similarity, but computed using surface word instead of POS tags.

6.5 Experiments

The experiments presented here aim to evaluate and compare the performance of different methods of training the translation model, including heuristics and the single-class classifier, first according to $\hat{P}P$ measure (Section 6.2.3), and second according to translation performance.

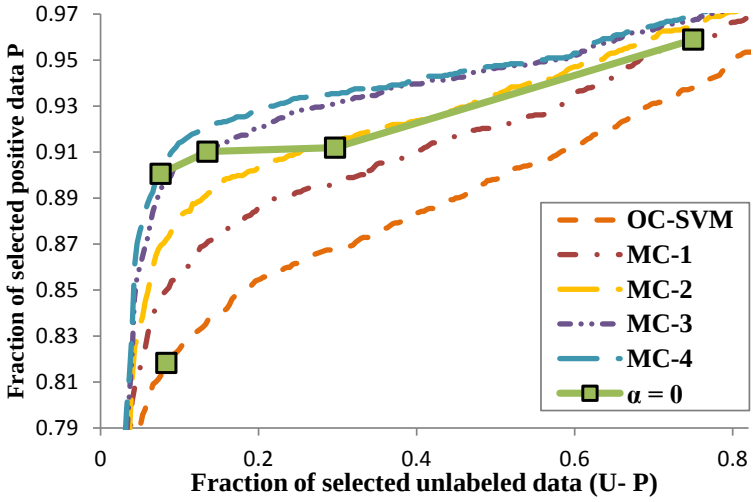


Figure 6.6: $\hat{P}P$ *measure: OC-SVM and MC-i, where i is the iteration.*

6.5.1 Data and Experimental Setup

For one-class SVM and the mapping convergence algorithm we use LIBSVM[4]. In our method we need to estimate a confidence measure of the classifier's output. Typically, we could use posterior probabilities as confidence scores but SVM does not have a direct probabilistic interpretation. Although standard SVMs do not produce such probabilities, they can be estimated using a method proposed in (Platt, 1999) that fits a logistic function to the output of an SVM. This algorithm assumes equal distribution of positive and negative examples in training and test sets. This is not the case in a one-class setting, nor in the convergence steps where the distribution in the training set actually changes on each step and converges to the actual one. Therefore, we slightly altered the code of LIBSVM so that it directly outputs the distance to the decision hyperplane which is used as a confidence measure of prediction.

For translation experiments we built several phrase-based, Arabic to English state-of-the-art SMT systems with Moses as described in the previous chapters. The same subsets of the NIST data are used for tuning and translation evaluation. We reconsider the same 30K sentences from the NIST'09 we used for the small task in previous chapter.

These sentences are input to the algorithm described in Section 6.1.2. U contains all possible phrase pairs with maximum phrase size of 3, for each of which we compute the set of features described in Section 6.4. We then use a phrase table built from U with both oracle decoders presented in Section 6.3 to decode a held-out parallel corpus of 2K sentences. Phrase pairs used by the decoder constitute positive examples, of which 80% are added to the training set P while the remaining are used for evaluation P_{test}.

6.5.2 Classification Performance: $\hat{P}P$

We use the positive examples in P to train a one-class SVM that is used to score all phrase pairs in U, of which the worst scoring 10% examples are considered strong negatives, and used to boost the MC algorithm. The parameters ν and γ of OC-SVM and all classifiers trained in MC iterations are tuned using grid search and cross-validation to maximize the $\hat{P}P$ measure.

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

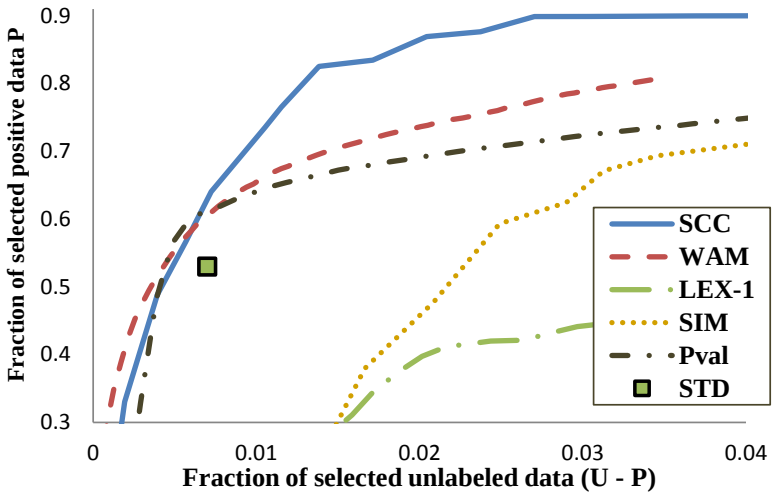


Figure 6.7: $\hat{P}P$ *measure: single class classifier (SCC) and some feature functions*

The performance of the OC-SVM and the binary classifiers resulting from subsequent iterations of MC are displayed in Figure 6.6. Each curve is obtained as described in Section 6.2.3, by quantizing the related classifier scores to divide the set $U - P$ into 300 quantiles of equal sizes and use the boundaries as different values for the threshold α.

Each point on the plot reflects on the y-axis the percentage of selected positive phrase pairs from P_{test}, against the selected percentage of $U - P$ on the x-axis.

The OC-SVM, depicted by the solid curve, achieves already a reasonable performance, identifying about 82% of positive examples while discarding about 90% of the rest.

Better percentages are achieved by subsequent iterations of MC, identifying 91% of positive examples and discarding 94% of the rest. The solid line connecting different points across curves plots the performance of the standard SVM classifiers at the threshold $\alpha = 0$.

Similarly to classifier scores, the different feature scores are quantized to obtain the curves depicted in Figure 6.7, where the curve corresponding to the best classifier is reproduced for comparison. We note that the single class classifier (SCC), which combines several features, achieves the best $\hat{P}P$ performance, improving on any feature acting solely.

6.5.3 Translation Performance: BLEU

6.5.3.1 Phrase pairs scoring method

We study in this section the translation performance in BLEU for each phrase pair selection score.

Similarly to the previous section, scores produced by the best classifier and different feature functions, are quantized into several quantiles per scoring method

Each corresponding threshold α is used to construct a phrase table by retaining all phrase pairs having a higher score and estimating standard models described in (Koehn, Och, and Marcu, 2003). After tuning the parameters of the translation systems[5] on the development corpus, BLEU is computed for each phrase table as the translation performance on the test corpus.

[5]In total we have 19 systems for the SCC classifier, 16 for the WAM feature, and 5 for each of the

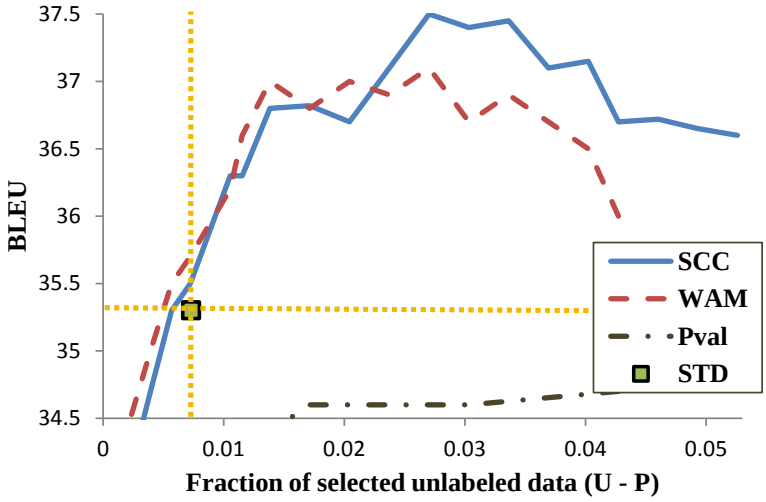


Figure 6.8: *BLEU: SCC and some feature functions*

Figure 6.8 plots BLEU as a function of the percentage of retained phrase pairs, which corresponds to the size of the phrase table. Figure 6.8 shows that for any given threshold, extraction based on the weighted alignment matrix (WAM) feature achieves the best performance in BLEU among features, with improvements over the standard baseline that ranges from slight to significant with different sizes of the phrase table. The SCC classifier achieves better BLEU score than WAM only for small values of α, which correspond to large phrase tables, while attaining comparable or slightly worst results for higher values of α (smaller phrase tables). Extraction based on scores by any other feature function, results in loss in performance. We note also that both the SCC classifier and WAM based approach performs better than the standard extraction for the same size of phrase table, and require fewer phrase pairs in order to obtain comparable performance, thus can be used for pruning large phrase tables.

6.5.3.2 Using additional phrase table features

We conducted an additional experiment where we incorporate the classifier score as an additional accuracy-based feature to the translation log-linear model and let MERT tune its weight.

Figure 6.10 shows, for different phrase table sizes, slight improvements in BLEU scores for systems that use this feature over the baselines that do not. Nevertheless, this feature is effective only for larger, noisier phrase tables. Similar behaviors are observed when adding all the other feature functions described in Section 6.4 are incorporated simultaneously[6].

6.5.4 Discussion

We would like to further analyze the dynamics of the scores computed with different methods. We consider three methods: the single-class classifier, the weighted alignment matrix and

remaining features

[6]We had to run MERT 3 times for each point and take the maximum BLEU score in this experiment since it was less stable when optimizing all the new features.

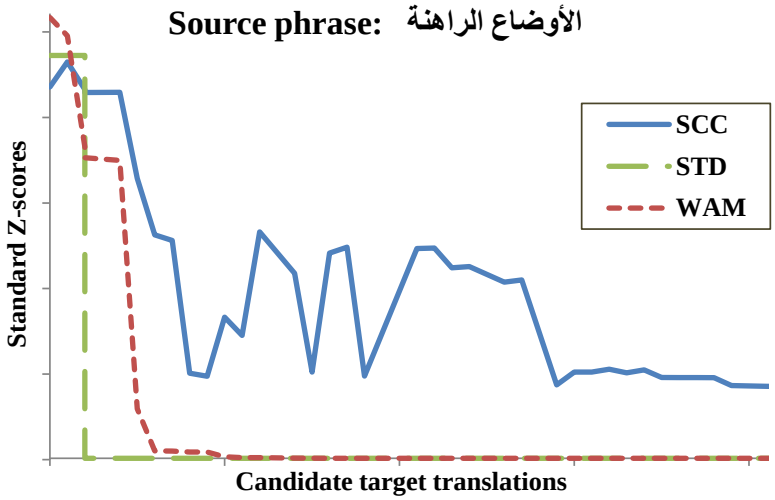


Figure 6.9: *Comparison of different phrase pair scoring methods. Z-scores of candidate phrase translations of the Arabic phrase "*الأوضاع الراهنة *(current situations)".*

the standard extraction heuristic. Figure 6.9 shows for each method and for a given source phrase, the score of all corresponding phrase pairs on the y-axis. The x-axis enumerate the target phrases in descending order of their scores for the standard extraction heuristic. Figure 6.9 reveals that substituting the standard heuristic scores with the weighted alignment matrix scores and further with the classifier score, has two effects: (1) it modifies the score and the rank of some phrase pairs causing the extraction of previously missed ones; (2) it smooths the scores and enhances the control over the selection process using the threshold α.

We note that while all of these three scoring methods identify well most of the best phrase pairs and rank them high in the list, they differ in their ability to rank phrase pairs of worst quality. While the scores based on the weighted alignment matrix may be sufficient to construct high precision phrase tables with the best phrase pairs, recall oriented phrase tables require more sophisticated decision procedures to retrieve good translations in the large set of candidates that are difficult to distinguish and ignored by standard methods.

6.6 Summary

In this chapter, we presented a procedure for both extraction and evaluation of phrase pairs. Similarly to the word alignment case, phrase pair extraction can be formulated as a binary classification problem, in which we decide for each phrase pair whether we keep it or not. The phrase pair consistency with the underlying word alignment is the standard criteria to perform this selection. We proposed to add many other cues to support the selection decisions and to recover from word alignment errors. These cues are represented as feature functions and combined in a statistical model, learned from data. Learning such a classification model requires both positive and negative examples, however the concept of a negative phrase pair example is ill-defined and are hard to obtain, which leaves us with only positive examples. We tackle the problem of learning only from positive examples with a single-class classification approach, capable of distinguishing one class, for which positive instances exist, from data in a universal set containing one or several classes, for which no sample is available. We proposed a phrase extraction procedure which uses this model to evaluate all candidate

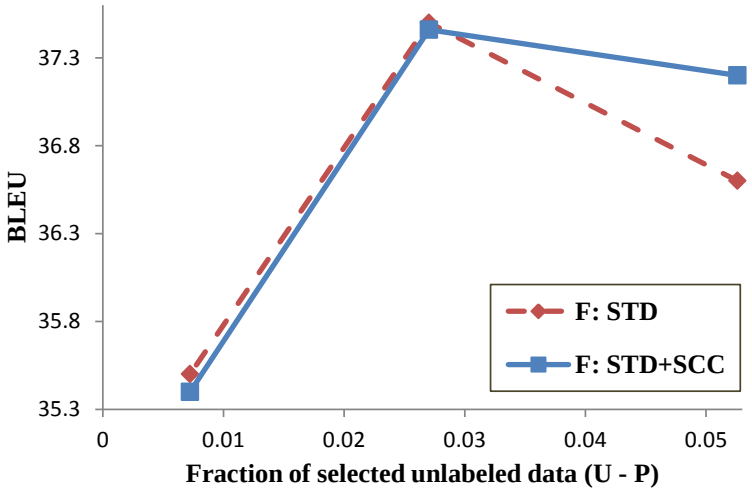


Figure 6.10: *BLEU: SCC scores as a new feature*

phrase-pairs, and uses a threshold on their score to decide on which phrase pairs to keep. The threshold parameter of this procedure is important since it allows to control the balance between precision and recall, and it can be tuned improved control over the size of the resulting phrase table, which is useful for fine tuning of the precision/recall balance.

The proposed method combines several features to produce fine-grained scores which helps exploring regions in the space of possible phrase pairs that are ignored by the standard extraction approach based on word alignment information only. This leads to improvements in BLEU scores for recall-oriented translation models, that are suitable for small training corpora. When large training corpora are available, precision-oriented translation models are preferred. The proposed procedure can be used in this setting by adjusting the value of the threshold to perform filtering. Future experiments are still needed to confirm this however. Additionally, we experimentally studied the effect on BLEU of adding new features to the phrase table and learning their weights with MERT, including a feature trained as a by-product of our procedure. This is helpful for recall-oriented phrase table to better recovery from noisy phrase pairs.

We conclude that extracting phrase pairs beyond what is obtained by the standard extraction heuristics is helpful when only a small amount of training data is available.

Conclusion

In this last chapter, we briefly summarize our contributions and discuss the conclusions that can be drawn from them. We also describe several ways of extending our work.

Contributions

We developed a maximum entropy framework for word alignment. In this framework, links are predicted independently from one another using a MaxEnt classifier. The interaction between alignment decisions is approximated using stacking techniques, which allows us to account for a part of the structural dependencies without increasing the complexity. This formulation can be seen as an alignment combination method, in which the union of several input alignments is used to guide the alignment. Additionally, they are used to compute a rich set of feature functions. We conducted a detailed study of the oracle AER that can be achieved for several combinations of input alignments. This is done by measuring the percentage of manual links found in their union. We obtained state of the art results in terms of alignment quality as measured by theAER and translation quality as measured by BLEU on large-scale Arabic-English NIST'09 systems. This model outperforms several baselines including the generative IBM and HMM models, and a discriminative CRF-based model. Several conclusions can be drawn from this work:

- Using a small amount of manually annotated data, large improvements in alignment quality can be achieved efficiently.
- A simple MaxEnt model which allows for exact and efficient inference may outperform its CRF counterpart which has more complex structure.
- A good estimation of the link posterior probabilities is very important. Thresholding these posteriors allows us to tune the trade-off between precision and recall to obtain the lowest AER. Furthermore, we can use phrase pair extraction procedures that use the posteriors instead of the "Viterbi" alignment. Doing this way, the entire alignment distribution is taken into consideration, which alleviates alignment errors and delivers better translation performance.
- The AER oracle study showed that almost all links annotated by humans are found in the neighborhood of IBM and HMM alignment. More importantly, the union of IBM1 and HMM is sufficient to identify the majority of correct alignments. The consequence is that we can dispense with more complex IBM4 alignments which significantly reduces the overall complexity, while achieving better alignment quality and improving translation performance compared to symmetrized IBM4 alignments.
- The standard symmetrization heuristic is important to improve the quality but it makes many errors that can be corrected with a simple model.
- The ℓ^1 regularization is very helpful. It performs feature selection and results in sparse and interpretable models while decreasing the AER.

- We bring a new light on the long standing debate about the correlation between alignment quality and translation. We show a high correlation between the quality BLEU and the AER, and even a higher correlation with the $F_{0.3}$ metric.

- Beside precision and recall, several alignment characteristics contribute to the translation quality. Finding a good balance between the alignment's precision and recall, its sparsity, and the number of unaligned words and extracted phrases, is necessary to enhance the translation quality.

- The phrase pairs extraction method based on the posteriors is very important since it provides a finer control of the size of the phrase table. This enables to find the best balance between phrase pairs precision and recall for the task at hand.

We have also presented a translation quality informed procedure for both extraction and evaluation of phrase pairs. Similar to the word alignment, we have reformulated the problem in the supervised framework in which we decide for each phrase pair whether we keep it or not. This offers a principled way to combine several features to make the procedure more robust to alignment difficulties. We use a simple and effective method to annotate phrase pairs that are useful for translation. Using machine learning techniques based on positive examples only these annotations can be used to learn phrase alignment decisions. Using this approach we obtain improvements in BLEU scores for recall-oriented translation models, that are suitable for small training corpora. Several conclusions can be drawn from this work:

- While posterior-based extraction procedures improve over the standard heuristic, they still suffer from alignment errors. Combining additional features helps recovering from such errors and improving the extraction.

- The discriminative model-based extraction, trained from positive examples only, produces better scores for phrase pairs than the heuristics. This helps exploring regions in the space of possible phrase pairs that are ignored by the standard approaches.

- When only small training corpora are available, recall-oriented translation models are preferred over precision-oriented, which are better for larger training corpora. Therefore, an extraction procedure that allows to control the balance between the quality of extracted phrase pairs and their coverage is very important.

Future Work

A main contribution of this dissertation is to show that only a small number of manually aligned sentences is sufficient to obtain large improvements in the alignment quality. This is very encouraging to look for more efficient way to benefit from human annotations. However, the word alignment may be ambiguous as in the case of idioms and expressions where several alignment patterns are possible. This ambiguity is typically dealt with by developing a style guide that is used to train annotators how to resolve ambiguity in a consistent manner. An interesting alternative would be to incorporate all such correct alignments into a model that permits learning from ambiguous labels. This is advantageous for at least two reasons. First, it is not required to make a commitment to an annotation style and no arbitrary disambiguation is needed since the ambiguity is marginalized during training. Second, instead of depending on an "expert" to obtain a "high quality" annotation, crowd-sourcing can be use to produce many redundant and possibly noisy annotations performed by "non-experts". This can drastically reduce the annotation cost. Several approaches to learn from ambiguous data can be considered. A very simple method is to learn a separate model from each set of annotation and combine the models *a posteriori*; or train the them jointly (Sutton, McCallum,

and Rohanimanesh, 2007). Alternatively, a single model can be learned using an objective that permits multiple references Dyer (2009).

On a related aspect, another major contribution of our work is to show that a simple model, which does not directly take the interactions between links into consideration, achieves a state of the art performance. The quality of link posterior estimates also improves. This is done using supervised discriminative learning techniques, and a rich set of features functions. Interestingly, these improvements were achieved at much lower computational cost than approaches that directly model the structure, and in which posterior estimation is not always tractable. An interesting line of research is to apply the same ideas to compute phrase pair posteriors, which are useful for applications such as SMT. Instead of computing such a confidence measure for phrase pairs using a heuristic or using a OC-SVM model, we may use the manual word alignment to define examples of good and bad phrase pairs and use them to learn a model of the phrase posterior. Since it is difficult to obtain full phrase pair annotations, it would be interesting to try also learning techniques from partial and possibly ambiguous annotations.

On the short term, it may be interesting to consider a few extensions to our work so as to answer some open questions. We draw our conclusions for the Arabic-English language pair. These conclusions are to be validated for additional language pairs and training data sets. Applying our alignment model to other applications than SMT is also interesting. Applications such as cross-lingual information retrieval and paraphrases may be more sensitive to improvements in alignment quality. Our word-based MaxEnt model is optimized using the likelihood objective. We are interested in studying the effect of using other objectives such as AER and the F-measure as done in several work in the literature. Our preliminary results using an approximation of the AER did not yield any improvements but further investigations are still to be performed. The MaxEnt model considers the links in the union of input alignments, which amounts to a small fraction of the entire alignment matrix, e. g. 3.3 % for the union of two IBM4 alignments. Adding the links in a window of size one around each union point increases this percentage to 15.7%. Adding point in more gradual fashion could help spotting good alignment points in the neighborhood without scanning large percentage of the matrix. We tried a heuristic similar to GDFA to add points to unaligned words, and we also tried random sampling but further experiments are needed to elaborate on this point. We showed that our discriminative extraction procedure is effective in producing recall-oriented phrase tables. It can also be used to perform filtering of large phrase tables by simply adjusting the value of the threshold. We sill need to conduct some experiments to test the performance in these settings. Both of our word-based and phrase-based alignment models rely on a threshold which is tuned to balance precision and recall. Currently we perform a grid search to select it optimal value. It would be interesting to consider other faster methods to induct the threshold from the properties of the task at hand.

Publications by the Author

2012

- Andrea Gesmundo and Nadi Tomeh.
  HadoopPerceptron: a Toolkit for Distributed Perceptron Training and Prediction with MapReduce. *13th Conference of the European Chapter of the Association for computational Linguistics - EACL'12, demo session.* Avignon, France. April 23-27, 2012.

2011

- Nadi Tomeh, Marco Turchi, Guillaume Wisniewski, Alexandre Allauzen and François Yvon.
  How Good Are Your Phrases? Assessing Phrase Quality with Single Class Classification. *International Workshop on Spoken Language Translation - IWSLT'11.* San Francisco, CA, USA. December 8-9, 2011.

- Nadi Tomeh, Alexandre Allauzen and François Yvon.
  Discriminative Weighted Alignment Matrices For Statistical Machine Translation. *The 15th Annual Conference of the European Association for Machine Translation - EAMT'11.* Leuven, Belgium. May 30-31, 2011.

- Nadi Tomeh, Alexandre Allauzen and François Yvon.
  Estimation d'un modèle de traduction à partir d'alignements mot-à-mot non-déterministes. Traitement Automatique des Langues Naturelles, TALN'2011. Montpellier, France. June 27-July 1, 2011.

- Nadi Tomeh, Alexandre Allauzen, Thomas Lavergne and François Yvon.
  Designing an Improved Discriminative Word Aligner. *International Journal of Computational Linguistics and Applications, IJCLA.* Tokyo, Japan. February 20-26, 2011.

2010

- Nadi Tomeh, Alexandre Allauzen, Guillaume Wisniewski and François Yvon.
  Refining Word Alignment with Discriminative Training. *The 9th Conference of the Association for Machine Translation in the Americas, AMTA'10.* Denver, Colorado, USA. October 31-November 5 2010.

2009

- Alexandre Allauzen, Josep M. Crego, Nadi Tomeh and François Yvon.
  The LIMSI Statistical Machine Translation System for NIST MT'09. *NIST Open Machine Translation 2009 Evaluation.* Ottawa, Ontario, Canada. August 31-Septembre 1, 2009.

- Nadi Tomeh, Nicola Cancedda and Marc Dymetman.
  Complexity-Based Phrase-Table Filtering for Statistical Machine Translation. *The 12th Machine Translation Summit, MT Summit XII.* Ottawa, Ontario, Canada. August 26-30, 2009.

2008

- Nadi Tomeh.
  Phrase-Table Filtering for a Statistical Machine Translation System. *Master Thesis, University Joseph Fourier.* Grenoble, France, June 2008.

Bibliography

(sorted by ascending date)

Dice, L. R. (07/1945). "Measures of the Amount of Ecologic Association Between Species". In: *Ecology* 26.3, pp. 297–302.

Vinay, Jean-Paul and Jean Darbelnet (1958). *Stylistique comparée du francais et de l'anglais: Méthode de Traduction*. Paris: Didier.

Jakobson, Roman (1959). "On Linguistic Aspects of Translation". In: *On Translation*. Ed. by LawrenceEditor Venuti. Vol. 3. Routledge, pp. 138–143. URL: http://www.stanford.edu/~eckert/PDF/jakobson.pdf.

Bar-Hillel, Y. (1964). "A demonstration of the nonfeasibility of fully automatic high quality machine translation." In: *Language and Information: Selected essays on their theory and application*. Jerusalem, Israel: The Jerusalem Academic Press Ltd., pp. 174–179.

Baum, L. E. and T. Petrie (1966). "Statistical inference for probabilistic functions of finite state Markov chains". In: *Annals of Mathematical Statistics* 37, pp. 1554–1563.

Viterbi, A (1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE Transactions on Information Theory* 13.2, pp. 260–269.

Lewis II, P. M. and R. E. Stearns (07/1968). "Syntax-Directed Transduction". In: *J. ACM* 15 (3), pp. 465–488. ISSN: 0004-5411. DOI: http://doi.acm.org/10.1145/321466.321477. URL: http://doi.acm.org/10.1145/321466.321477.

Aho, Alfred V. and Jeffrey D. Ullman (1969). "Syntax Directed Translations and the Pushdown Assembler". In: *J. Comput. Syst. Sci.* 3.1, pp. 37–56.

Baum, Leonard E., Ted Petrie, George Soules, and Norman Weiss (1970). "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains". In: *The Annals of Mathematical Statistics* 41.1, pp. 164–171. URL: http://dx.doi.org/10.2307/2239727.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society, Series B* 39.1, pp. 1–38.

Valiant, L.G. (1979). "The complexity of computing the permanent". In: *Theoretical Computer Science* 8.2, pp. 189 –201. ISSN: 0304-3975. DOI: 10.1016/0304-3975(79)90044-6. URL: http://www.sciencedirect.com/science/article/pii/0304397579900446.

Ricoeur, P. and J.B. Thompson (1981). *Hermeneutics and the Human Sciences: Essays on Language, Action, and Interpretation*. Cambridge University Press. ISBN: 9780521280020. URL: http://books.google.fr/books?id=8L1CwJ0U_DsC.

Nagao, Makoto (1984). "A framework of a mechanical translation between Japanese and English by analogy principle". In: *Proc. of the international NATO symposium on Artificial and human intelligence*. Lyon, France: Elsevier North-Holland, Inc., pp. 173–180. ISBN: 0-444-86545-4. URL: http://dl.acm.org/citation.cfm?id=2927.2938.

Pearl, Judea (1984). *Heuristics - intelligent search strategies for computer problem solving*. Addison-Wesley series in artificial intelligence. Addison-Wesley, pp. I–XVII, 1–382. ISBN: 978-0-201-05594-8.

Newmark, Peter (1988). *A Textbook of Translation*. Prentice Hall. ISBN: 0139125930.

Liu, D. C. and J. Nocedal (12/1989). "On the limited memory BFGS method for large scale optimization". In: *Math. Program.* 45.3, pp. 503–528. ISSN: 0025-5610. DOI: 10.1007/BF01589116. URL: http://dx.doi.org/10.1007/BF01589116.

Klavans, Judith and Evelyne Tzoukermann (1990). "The BICORD system: combining lexical infor-

mation from bilingual corpora and machine readable dictionaries". In: *Proceedings of the 13th conference on Computational linguistics - Volume 3*. COLING '90. Helsinki, Finland: Association for Computational Linguistics, pp. 174–179. ISBN: 952-90-2028-7. DOI: http://dx.doi.org/10.3115/991146.991177. URL: http://dx.doi.org/10.3115/991146.991177.

Lari, K. and S. J. Young (1990). "The estimation of stochastic context-free grammars using the Inside-Outside algorithm". In: *Computer Speech and Language* 4, pp. 35–56.

Brown, Peter F., Jennifer C. Lai, and Robert L. Mercer (1991). "ALIGNING SENTENCES IN PARALLEL CORPORA". In: *Proceedings of the 29th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Gale, William A. and Kenneth W. Church (1991). "A program for aligning sentences in bilingual corpora". In: *Proceedings of the 29th annual meeting on Association for Computational Linguistics*. ACL '91. Berkeley, California: Association for Computational Linguistics, pp. 177–184. DOI: 10.3115/981344.981367. URL: http://dx.doi.org/10.3115/981344.981367.

Warwick, Susan and G Russell (1992). "Bilingual Concordancing and Bilingual Lexicography". In: *EURALEX 4th International Congress*. Ed. by Editors.

Wolpert, D. H. (1992). "Original Contribution: Stacked generalization". In: *Neural Netw.* 5.2, pp. 241–259. ISSN: 0893-6080. DOI: http://dx.doi.org/10.1016/S0893-6080(05)80023-1.

Brown, Peter F., Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer (1993). "The mathematics of statistical machine translation: parameter estimation". In: *Comput. Linguist.* 19.2, pp. 263–311.

Chen, Stanley F. (1993). "ALIGNING SENTENCES IN BILINGUAL CORPORA USING LEXICAL INFORMATION". In: *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dagan, Ido, Kenneth W. Church, and William A. Gale (1993). "Robust Bilingual Word Alignment for Machine Aided Translation". In: *In Proceedings of the Workshop on Very Large Corpora*, pp. 1–8.

Dunning, Ted (03/1993). "Accurate methods for the statistics of surprise and coincidence". In: *Comput. Linguist.* 19.1, pp. 61–74. ISSN: 0891-2017. URL: http://dl.acm.org/citation.cfm?id=972450.972454.

Gale, William A. and Kenneth Ward Church (1993). "A program for aligning sentences in bilingual corpora". In: *Computational Linguistics* 19.1.

Kay, Martin and Martin Röscheisen (1993). "Text-Translation Alignment". In: *Computational Linguistics* 19.1.

Macklovitch, Elliott (10/1994). "Using Bi-textual Alignment for Translation Validation: the TransCheck System". In: *Actes du First Conference of the Association for Machine Translation in the Americas (AMTA-94)*. Columbia, É-U.

Utsuro, Takehito, Hiroshi Ikeda, Masaya Yamane, Yuji Matsumoto, and Makoto Nagao (1994). "Bilingual Text, Matching using Bilingual Dictionary and Statistics". In: *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*.

Wu, Dekai (1994). "ALIGNING A PARALLEL ENGLISH-CHINESE CORPUS STATISTICALLY WITH LEXICAL CRITERIA". In: *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dougherty, James, Ron Kohavi, and Mehran Sahami (1995). "Supervised and Unsupervised Discretization of Continuous Features". In: *ICML*, pp. 194–202.

Gaussier, Éric and Jean-Marc Langé (1995). "Modèles statistiques pour l'extraction de lexiques bilingues". In: *Traitement Automatique des Langues (TAL)*, pp. 133–155.

Ostendorf, M., V. Digalakis, and O. A. Kimball (1995). "From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition". In: *IEEE Transactions on Speech and Audio Processing* 4, pp. 360–378.

Wu, D. and X. Xia (1995). "Large-scale automatic extraction of an English-Chinese lexicon". In: *Machine Translation* 9.3–4, pp. 285–313.

Wu, Dekai (06/1995a). "An Algorithm for Simultaneously Bracketing Parallel Texts by Aligning Words". In: *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, pp. 244–251. DOI: 10.3115/981658.981691. URL: http://www.aclweb.org/anthology/P95-1033.

Wu, Dekai (1995b). "Trainable Coarse Bilingual Grammars for Parallel Text Bracketing". In: *Proceedings of the Third Workshop on Very Large Corpora (VLC)*.

Berger, Adam L., Vincent J. Della Pietra, and Stephen A. Della Pietra (03/1996). "A max-

imum entropy approach to natural language processing". In: *Comput. Linguist.* 22 (1), pp. 39–71. ISSN: 0891-2017.

Chen, Stanley F. and Joshua Goodman (1996). "An empirical study of smoothing techniques for language modeling". In: *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. ACL '96. Santa Cruz, California: Association for Computational Linguistics, pp. 310–318. DOI: http://dx.doi.org/10.3115/981863.981904. URL: http://dx.doi.org/10.3115/981863.981904.

Melamed, I. Dan (1996a). "A Geometric Approach to Mapping Bitext Correspondence". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Melamed, I. Dan (1996b). "Automatic construction of clean broad-coverage translation lexicons". In: *Proceedings of the Conference of the Association for Machine Translation in the Americas*.

Smadja, Frank, Kathleen R. McKeown, and Vasileios Hatzivassiloglou (03/1996). "Translating collocations for bilingual lexicons: a statistical approach". In: *Comput. Linguist.* 22.1, pp. 1–38. ISSN: 0891-2017. URL: http://dl.acm.org/citation.cfm?id=234285.234287.

Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the Lasso". In: *J.R.Statist.Soc.B* 58.1, pp. 267–288.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann (1996). "HMM-Based Word Alignment in Statistical Translation". In: *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*.

Chang, Jason S. and Mathis H. Chen (1997a). "An Alignment Method for Noisy Parallel Corpora based on Image Processing Techniques". In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chang, Jason S. and Mathis H. Chen (1997b). "An alignment method for noisy parallel corpora based on image processing techniques". In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. ACL '98. Madrid, Spain: Association for Computational Linguistics, pp. 297–304. DOI: 10.3115/976909.979655. URL: http://dx.doi.org/10.3115/976909.979655.

Ker, Sue J. and Jason S. Chang (06/1997). "A class-based approach to word alignment". In: *Comput. Linguist.* 23.2, pp. 313–343. ISSN: 0891-2017. URL: http://dl.acm.org/citation.cfm?id=972695.972699.

Langlais, Philippe (1997). "Alignement de corpus bilingues: intérêts, algorithmes et évaluations". In: *1st International Conference on Natural Language Processins (FracTAL)*. publié dans *Bulletin de Linguistique Appliquée et Générale*. Université de Franche-Comté, France, pp. 245–254.

Melamed, I. Dan (1997). "A Portable Algorithm for Mapping Bitext Correspondence". In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mohri, Mehryar (06/1997). "Finite-State Transducers in Language and Speech Processing". In: *Computational Linguistics* 23.2, pp. 269–311.

Resnik, Philip and David Yarowsky (1997). "A Perspective on Word Sense Disambiguation Methods and Their Evaluation". In: pp. 79–86.

Shuttleworth, Mark and Moira Cowie (1997). *Dictionary of translation studies*. 98209732 Mark Shuttleworth & Moira Cowie. 22 cm. Includes bibliographical references (p. -233). Manchester, UK: St. Jerome Pub.

Wang, Ye-Yi and Alex Waibel (1997). "Decoding Algorithm in Statistical Machine Translation". In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wu, Dekai (1997). "Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora". In: *Computational Linguistics* 23.3.

Knight, Kevin and Yaser Al-Onaizan (1998). "Translation with Finite-State Devices". In: *Proceedings of the Third Conference of the Association for Machine Translation in the Americas on Machine Translation and the Information Soup*. AMTA '98. London, UK, UK: Springer-Verlag, pp. 421–437. ISBN: 3-540-65259-0. URL: http://dl.acm.org/citation.cfm?id=648179.749225.

Langlais, Philippe, Michel Simard, and Jean Veronis (1998). "Methods and Practical Issues in Evaluating Alignment Techniques". In: *Proceedings of the 36th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Langlais, Philippe and Jean Véronis (04/1998). "Progress in parallel text alignment techniques for multilingual lexical acquisition: the ARCADE evaluation exercise". In: *2nd Workshop on Lexical Semantics Systems (WLSS'98)*. Pisa, Italy.

Melamed, I. Dan (1998). "Annotation Style Guide for the Blinker Project". In: *CoRR*.

Goodman, Joshua (12/1999). "Semiring parsing". In: *Comput. Linguist.* 25.4, pp. 573–605. ISSN: 0891-2017. URL: http://dl.acm.org/citation.cfm?id=973226.973230.

Knight, Kevin (12/1999). "Decoding complexity in word-replacement translation models". In: *Comput. Linguist.* 25.4, pp. 607–615. ISSN: 0891-2017. URL: http://dl.acm.org/citation.cfm?id=973226.973232.

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Melamed, I. Dan (1999). "Bitext Maps and Alignment via Pattern Recognition". In: *Computational Linguistics* 25.1, pp. 107–130.

Och, Franz Josef, Christoph Tillmann, and Hermann Ney (1999). "Improved Alignment Models for Statistical Machine Translation". In: *Proceedings of the Joint Conference of Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC)*, pp. 20–28.

Platt, John C. (1999). "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". In: *Advances in Large-Margin Classifiers*. MIT Press, pp. 61–74.

Resnik, Philip (1999). "Mining the Web for Bilingual Text". In: *ACL*.

Tax, David M. J. and Robert P. W. Duin (11/1999). "Support vector domain description". In: *Pattern Recogn. Lett.* 20 (11-13), pp. 1191–1199. ISSN: 0167-8655. DOI: http://dx.doi.org/10.1016/S0167-8655(99)00087-2. URL: http://dx.doi.org/10.1016/S0167-8655(99)00087-2.

Tiedemann, Jörg (1999). "Word alignment - step by step". In: *Proceedings of the 12th Nordic Conference on Computational Linguistics (NODALIDA)*. University of Trondheim, Norway, pp. 216–227.

Ahrenberg, Lars, Magnus Merkel, Anna Sågvall Hein, and Jörg Tiedemann (2000). *Evaluation of Word Alignment Systems*.

Cristianini, Nello and John Shawe-Taylor (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. 1st ed. Cambridge University Press.

Langlais, Philippe, George Foster, and Guy Lapalme (05/2000). "Transtype: a Computer-Aided Translation Typing System". In: *Satellite Workshop of NAACL/ANLP*. Seattle, Washington, USA, pp. 46–51.

Melamed, I. Dan (06/2000). "Models of translational equivalence among words". In: *Comput. Linguist.* 26 (2), pp. 221–249. ISSN: 0891-2017.

Nigam, Kamal, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell (2000). "Text Classification from Labeled and Unlabeled Documents using EM". In: *Machine Learning* 39.2/3, pp. 103–134.

Schölkopf, Bernhard, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett (05/2000). "New Support Vector Algorithms". In: *Neural Comput.* 12 (5), pp. 1207–1245. ISSN: 0899-7667. DOI: 10.1162/089976600300015565. URL: http://dl.acm.org/citation.cfm?id=1139689.1139691.

Véronis, Jean, ed. (2000). *Parallel Text Processing: Alignment and Use of Translation Corpora (Text, Speech and Language Technology)*. Springer. ISBN: 0792365461.

Germann, Ulrich, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada (2001). "Fast Decoding and Optimal Decoding for Machine Translation". In: *Proceedings of the 39th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 282–289. ISBN: 1-55860-778-1. URL: http://dl.acm.org/citation.cfm?id=645530.655813.

Schölkopf, Bernhard, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson (07/2001). "Estimating the Support of a High-Dimensional Distribution". In: *Neural Comput.* 13 (7), pp. 1443–1471. ISSN: 0899-7667. DOI: 10.1162/089976601750264965. URL: http://portal.acm.org/citation.cfm?id=1119748.1119749.

Tax, D.M.J. (06/2001). "One-class classification". phd. Delft: Delft University of Technology.

Varea, Ismael García, Franz J. Och, Hermann Ney, and Francisco Casacuberta (2001). "Refined lexicon models for statistical machine translation using a maximum entropy approach". In: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. ACL '01. Toulouse, France: Association for Computational Linguistics, pp. 204–211. DOI: http:

//dx.doi.org/10.3115/1073012.1073039. URL: http://dx.doi.org/10.3115/1073012.1073039.

Yamada, Kenji and Kevin Knight (2001). "A Syntax-Based Statistical Translation Model". In: *Proceedings of the 39th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Yarowsky, David, Grace Ngai, and Richard Wicentowski (2001). "Inducing multilingual text analysis tools via robust projection across aligned corpora". In: *Proceedings of the first international conference on Human language technology research*. HLT '01. San Diego: Association for Computational Linguistics, pp. 1–8.

Collins, Michael (2002). "Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*. EMNLP '02. Association for Computational Linguistics, pp. 1–8. DOI: http://dx.doi.org/10.3115/1118693.1118694. URL: http://dx.doi.org/10.3115/1118693.1118694.

Davis, P.C. (2002). *Stone soup translation: the linked automata model*. Ohio State University. URL: http://books.google.com/books?id=385htwAACAAJ.

Doddington, George (2002). "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics". In: *Proceedings of the second international conference on Human Language Technology Research*. HLT '02. San Diego, California: Morgan Kaufmann Publishers Inc., pp. 138–145. URL: http://dl.acm.org/citation.cfm?id=1289189.1289273.

Koehn, Philipp and Kevin Knight (2002). "Learning a translation lexicon from monolingual corpora". In: *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition - Volume 9*. ULA '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 9–16. DOI: 10.3115/1118627.1118629. URL: http://dx.doi.org/10.3115/1118627.1118629.

Kueng, Tz-Liang and Keh-Yih Su (2002). "A Robust Cross-Style Bilingual Sentence Alignment Model". In: *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Kumar, Shankar and William Byrne (07/2002). "Minimum Bayes-Risk Word Alignments of Bilingual Texts". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Philadelphia: Association for Computational Linguistics, pp. 140–147.

Kwong, Oi Yee, Benjamin K. Tsou, Tom B. Y. Lai, Lawrence Y. L. Cheung, Francis C. Y. Chik, and Robert W. P. Luk (2002). "Alignment and extraction of bilingual legal terminology from context profiles". In: *COLING-02 on COMPUTERM 2002: second international workshop on computational terminology - Volume 14*. COMPUTERM '02. Association for Computational Linguistics, pp. 1–7. DOI: 10.3115/1118771.1118775. URL: http://dx.doi.org/10.3115/1118771.1118775.

Marcu, Daniel and William Wong (2002). "A phrase-based, joint probability model for statistical machine translation". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*. EMNLP '02. Morristown, NJ, USA: Association for Computational Linguistics, pp. 133–139.

Mohri, Mehryar, Fernando Pereira, and Michael Riley (2002). "Weighted finite-state transducers in speech recognition". In: *Computer Speech & Language* 16.1, pp. 69–88.

Moore, Robert C. (2002). "Fast and Accurate Sentence Alignment of Bilingual Corpora". In: *Machine Translation: From Research to Real Users, 5th Conference of the Association for Machine Translation in the Americas, AMTA 2002 Tiburon, CA, USA, October 6-12, 2002, Proceedings*. Ed. by Stephen D. Richardson. Vol. 2499. Lecture Notes in Computer Science. Springer. ISBN: 3-540-44282-0.

Nivre, Joakim (2002). "On Statistical Methods in Natural Language Processing". In: *Promote IT. Second Conference for the Promotion of Research in IT at New Universities and University Colleges in Sweden*.

Och, Franz Josef and Hermann Ney (2002). "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation". In: *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 295–302.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). "BLEU: a method for automatic evaluation of machine translation". In: *Proc. of the 40th Annual Meeting on ACL*. Philadelphia, Pennsylvania, pp. 311–318.

Toutanova, Kristina, H. Tolga Ilhan, and Christopher D. Manning (2002). "Extensions to HMM-based statistical word alignment models". In:

Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10. EMNLP '02. Association for Computational Linguistics, pp. 87–94.

Xu, Jinxi, Alexander Fraser, and Ralph Weischedel (2002). "Empirical studies in strategies for Arabic retrieval". In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. SIGIR '02. Tampere, Finland: ACM, pp. 269–274. ISBN: 1-58113-561-0. DOI: 10.1145/564376.564424. URL: http://doi.acm.org/10.1145/564376.564424.

Zens, Richard, Franz Josef Och, and Hermann Ney (2002). "Phrase-Based Statistical Machine Translation". In: *Proceedings of the German Conference on Artificial Intelligence (KI 2002)*.

Cherry, Colin and Dekang Lin (2003). "A Probability Model to Improve Word Alignment". In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Ed. by Erhard Hinrichs and Dan Roth, pp. 88–95. URL: http://www.aclweb.org/anthology/P03-1012.pdf.

Germann, Ulrich (2003). "Greedy Decoding for Statistical Machine Translation in Almost Linear Time". In: *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computation Linguistics (HLT-NAACL)*.

Gildea, Daniel (2003). "Loosly Tree-Based Alignment for Machine Translation". In: *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL)*.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu (2003). "Statistical phrase-based translation". In: *Proc. NAACL-HLT 2003*. Edmonton, Canada, pp. 48–54.

Kumar, Shankar and William Byrne (2003). "A Weighted Finite State Transducer Implementation of the Alignment Template Model for Statistical Machine Translation". In: *HLT-NAACL 2003: Main Proceedings*. Ed. by Marti Hearst and Mari Ostendorf. Edmonton, Alberta, Canada: Association for Computational Linguistics, pp. 142–149.

Nida, Eugene A. and Charles R. Taber (2003). *The Theory and Practice of Translation*. Brill Academic Pub. ISBN: 9004132813.

Och, Franz Josef (2003). "Minimum Error Rate Training in Statistical Machine Translation". In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Ed. by Erhard Hinrichs and Dan Roth, pp. 160–167. URL: http://www.aclweb.org/anthology/P03-1021.pdf.

Och, Franz Josef and Herman Ney (2003). "A systematic comparison of various statistical alignment models". In: *Comput. Linguist.* 29 (1), pp. 19–51.

Pang, Bo, Kevin Knight, and Daniel Marcu (2003). "Syntax-based alignment of multiple translations: extracting paraphrases and generating new sentences". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*. NAACL '03. Edmonton, Canada: Association for Computational Linguistics, pp. 102–109. DOI: 10.3115/1073445.1073469. URL: http://dx.doi.org/10.3115/1073445.1073469.

Resnik, Philip and Noah A. Smith (09/2003). "The Web as a parallel corpus". In: *Comput. Linguist.* 29.3, pp. 349–380. ISSN: 0891-2017. DOI: 10.1162/089120103322711578. URL: http://dx.doi.org/10.1162/089120103322711578.

Tiedemann, Jörg (2003a). "Recycling Translations – Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing". Anna Sågvall Hein, Åke Viberg (eds): Studia Linguistica Upsaliensia. PhD thesis. Uppsala, Sweden: Uppsala University. URL: http://stp.ling.uu.se/~joerg/phd/.

Tiedemann, Jörg (2003b). "Combining Clues for Word Alignment". In: *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL)*.

Venugopal, Ashish, Stephan Vogel, and Alex Waibel (2003). "Effective Phrase Translation Extraction from Alignment Models". In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Ed. by Erhard Hinrichs and Dan Roth, pp. 319–326. URL: http://www.aclweb.org/anthology/P03-1041.pdf.

Zens, Richard and Hermann Ney (07/2003). "A Comparative Study on Reordering Constraints in Statistical Machine Translation". In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, pp. 144–151. DOI: 10.3115/1075096.1075115. URL: http://www.aclweb.org/anthology/P03-1019.

Diab, Mona (2004). "Feasibility of Bootstrapping an Arabic WordNet Leveraging Parallel Corpora and an English WordNet". In: *English*.

Fung, Pascale and Percy Cheung (07/2004a). "Mining Very-Non-Parallel Corpora: Parallel Sentence and Lexicon Extraction via Bootstrapping and EM". In: *Proceedings of EMNLP 2004*. Ed. by Dekang Lin and Dekai Wu. Barcelona, Spain: Association for Computational Linguistics, pp. 57–63.

Fung, Pascale and Percy Cheung (08/2004b). "Multi-level Bootstrapping For Extracting Parallel Sentences From a Quasi-Comparable Corpus". In: *Proceedings of Coling 2004*. Ed. by. Geneva, Switzerland: COLING, pp. 1051–1057.

Galley, Michel, Mark Hopkins, Kevin Knight, and Daniel Marcu (2004). "What's in a translation rule?" In: *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Goutte, Cyril, Kenji Yamada, and Eric Gaussier (07/2004). "Aligning words using matrix factorisation". In: *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*. Barcelona, Spain, pp. 502–509.

Koehn, Philipp (2004). "Pharaoh: a beam search decoder for phrase-based statistical machine translation models". In: *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (AMTA 2004)*, pp. 115–124.

Kumar, Shankar and William Byrne (2004). "Minimum Bayes-Risk Decoding for Statistical Machine Translation". In: *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Matusov, Evgeny, Richard Zens, and Hermann Ney (2004). "Symmetric Word Alignments for Statistical Machine Translation". In: *Proceedings of Coling 2004*. Ed. by. Geneva, Switzerland: COLING, pp. 219–225.

Melamed, I. Dan (07/2004). "Statistical Machine Translation by Parsing". In: *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*. Barcelona, Spain, pp. 653–660.

Moore, Robert C. (07/2004a). "Improving IBM Word Alignment Model 1". In: *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*. Barcelona, Spain, pp. 518–525. DOI: 10.3115/1218955.1219021. URL: http://www.aclweb.org/anthology/P04-1066.

Moore, Robert C. (2004b). "On Log-Likelihood-Ratios and the Significance of Rare Events". In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. URL: http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Moore.pdf.

Och, Franz Josef and Hermann Ney (2004). "The Alignment Template Approach to Statistical Machine Translation". In: *Computational Linguistics* 30.4.

Och, Franz Josef, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alexander Fraser, Shankar Kumar, Libin Shen, David A. Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev (2004). "A Smorgasbord of Features for Statistical Machine Translation". In: *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Quirk, Chris, Chris Brockett, and William B. Dolan (2004). "Monolingual Machine Translation for Paraphrase Generation". In: *EMNLP*, pp. 142–149.

Raskutti, Bhavani and Adam Kowalczyk (06/2004). "Extreme re-balancing for SVMs: a case study". In: *SIGKDD Explor. Newsl.* 6 (1), pp. 60–69. ISSN: 1931-0145. DOI: http://doi.acm.org/10.1145/1007730.1007739. URL: http://doi.acm.org/10.1145/1007730.1007739.

Shen, Libin, Anoop Sarkar, and Franz Josef Och (2004). "Discriminative Reranking for Machine Translation". In: *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Smith, David A. and Noah A. Smith (2004). "Bilingual Parsing with Factored Estimation: Using English to Parse Korean". In: *EMNLP*, pp. 49–56.

Taskar, Ben (2004). "Learning structured prediction models: a large margin approach". PhD thesis. Stanford University. Computer Science Dept. CA. URL: http://ai.stanford.edu/\~{}btaskar/pubs/thesis.pdf.

Tiedemann, Jörg (2004). "Word to word alignment strategies". In: *Proceedings of Coling 2004*. Ed.

by. Geneva, Switzerland: COLING, pp. 212–218.

Tillmann, Christoph (2004). "A Unigram Orientation Model for Statistical Machine Translation". In: *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Zens, Richard and Hermann Ney (2004). "Improvements in Phrase-Based Statistical Machine Translation". In: *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Ayan, Necip Fazil, Bonnie J. Dorr, and Christof Monz (10/2005). "NeurAlign: Combining Word Alignments Using Neural Networks". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 65–72. URL: http://www.aclweb.org/anthology/H/H05/H05-1009.

Banerjee, Satanjeev and Alon Lavie (06/2005). "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 65–72. URL: http://www.aclweb.org/anthology/W/W05/W05-0909.

Bannard, Colin and Chris Callison-Burch (2005). "Paraphrasing with bilingual parallel corpora". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. ACL '05. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 597–604. DOI: 10.3115/1219840.1219914. URL: http://dx.doi.org/10.3115/1219840.1219914.

Chiang, David (06/2005). "A Hierarchical Phrase-Based Model for Statistical Machine Translation". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 263–270. URL: http://www.aclweb.org/anthology/P/P05/P05-1033.

Cohen, W. W. and V. R. Carvalho (2005). "Stacked sequential learning". In: *IJCAI*. Edinburgh, Scotland: Morgan Kaufmann Publishers Inc., pp. 671–676.

Deng, Yonggang and William Byrne (10/2005). "HMM Word and Phrase Alignment for Statistical Machine Translation". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 169–176. URL: http://www.aclweb.org/anthology/H/H05/H05-1022.

García-Varea, Ismael, Daniel Ortiz, Francisco Nevado, Pedro A. Gómez, and Francisco Casacuberta (2005). "Automatic Segmentation of Bilingual Corpora: A Comparison of Different Techniques". In: *IbPRIA (2)*, pp. 614–621.

Hatim, Basil A and Jeremy Munday (2005). *Translation: An Advanced Resource Book (Routledge Applied Linguistics)*. Routledge.

Hwa, Rebecca, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak (09/2005). "Bootstrapping parsers via syntactic projection across parallel texts". In: *Nat. Lang. Eng.* 11 (3), pp. 311–325. ISSN: 1351-3249.

Ittycheriah, Abraham and Salim Roukos (10/2005). "A Maximum Entropy Word Aligner for Arabic-English Machine Translation". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 89–96. URL: http://www.aclweb.org/anthology/H/H05/H05-1012.

Knight, Kevin and Daniel Marcu (2005). "Machine translation in the year 2004". In: *In Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP*. IEEE Computer Society, pp. 965–968.

Koehn, Philipp (09/2005). "Europarl: A Parallel Corpus for Statistical Machine Translation". In: *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*. Phuket, Thailand.

Koehn, Philipp, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot (10/2005). "Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation". In: *Proc. of the International Workshop on Spoken Language Translation*. Pittsburgh, PA, USA.

Liu, Yang, Qun Liu, and Shouxun Lin (06/2005). "Log-Linear Models for Word Alignment". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 459–466. URL: http:

//www.aclweb.org/anthology/P/P05/P05-1057.

Lopez, Adam and Philip Resnik (2005). "Improved HMM alignment models for languages with scarce resources". In: *Proceedings of the ACL Workshop on Building and Using Parallel Texts*. ParaText '05. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 83–86. URL: http://dl.acm.org/citation.cfm?id=1654449.1654464.

Moore, Robert C. (10/2005). "A Discriminative Framework for Bilingual Word Alignment". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 81–88. URL: http://www.aclweb.org/anthology/H/H05/H05-1011.

Simard, Michel, Nicola Cancedda, Bruno Cavestro, Marc Dymetman, Eric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlais, and Arne Mauser (2005). "Translating with non-contiguous phrases". In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. HLT '05. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 755–762.

Singh, Anil Kumar and Samar Husain (06/2005). "Comparison, Selection and Use of Sentence Alignment Algorithms for New Language Pairs". In: *Proceedings of the ACL Workshop on Building and Using Parallel Texts*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 99–106. URL: http://www.aclweb.org/anthology/W/W05/W05-0816.

Taskar, Ben, Simon Lacoste-Julien, and Dan Klein (10/2005). "A Discriminative Matching Approach to Word Alignment". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 73–80. URL: http://www.aclweb.org/anthology/H/H05/H05-1010.

Vogel, Stephan (09/2005). "PESA: Phrase Pair Extraction as Sentence Splitting". In: *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*. Phuket, Thailand.

Wang, Jianqiang (2005). "Matching meaning for cross-language information retrieval". PhD thesis. College Park, MD, USA.

Wu, Hua and Haifeng Wang (09/2005). "Boosting Statistical Word Alignment". In: *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*. Phuket, Thailand.

Yu, Hwanjo (11/2005). "Single-Class Classification with Mapping Convergence". In: *Mach. Learn.* 61 (1-3), pp. 49–69. ISSN: 0885-6125. DOI: 10.1007/s10994-005-1122-7. URL: http://portal.acm.org/citation.cfm?id=1108759.1108762.

Zhang, Hao and Daniel Gildea (2005). "Stochastic lexicalized inversion transduction grammar for alignment". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. ACL '05. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 475–482. DOI: http://dx.doi.org/10.3115/1219840.1219899. URL: http://dx.doi.org/10.3115/1219840.1219899.

Zou, Hui and Trevor Hastie (2005). "Regularization and variable selection via the Elastic Net". In: *Journal of the Royal Statistical Society, Series B* 67, pp. 573–580.

Ayan, Necip Fazil and Bonnie J. Dorr (06/2006a). "A Maximum Entropy Approach to Combining Word Alignments". In: *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. New York City, USA: Association for Computational Linguistics, pp. 96–103. URL: http://www.aclweb.org/anthology/N/N06/N06-1013.

Ayan, Necip Fazil and Bonnie J. Dorr (07/2006b). "Going Beyond AER: An Extensive Analysis of Word Alignments and Their Impact on MT". In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, pp. 9–16. URL: http://www.aclweb.org/anthology/P/P06/P06-1002.

Birch, Alexandra, Chris Callison-Burch, Miles Osborne, and Philipp Koehn (2006). "Constraining the phrase-based, joint probability statistical translation model". In: *Proceedings of the Workshop on Statistical Machine Translation*. StatMT '06. New York City, New York: Association for Computational Linguistics, pp. 154–157. URL: http://portal.acm.org/citation.cfm?id=1654650.1654675.

Blunsom, Phil and Trevor Cohn (07/2006). "Discriminative Word Alignment with Conditional Random Fields". In: *Proceedings of the 21st International Conference on Computational Lin-*

guistics and 44th Annual Meeting of the Association for Computational Linguistics. Sydney, Australia: Association for Computational Linguistics, pp. 65–72. URL: http://www.aclweb.org/anthology/P/P06/P06-1009.

Cherry, Colin and Dekang Lin (04/2006a). "A Comparison of Syntactically Motivated Word Alignment Spaces". In: *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento, Italy.

Cherry, Colin and Dekang Lin (07/2006b). "Soft Syntactic Constraints for Word Alignment through Discriminative Training". In: *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Sydney, Australia: Association for Computational Linguistics, pp. 105–112. URL: http://www.aclweb.org/anthology/P/P06/P06-2014.

Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer (12/2006). "Online Passive-Aggressive Algorithms". In: *J. Mach. Learn. Res.* 7, pp. 551–585. ISSN: 1532-4435. URL: http://portal.acm.org/citation.cfm?id=1248547.1248566.

Crego, Josep and José Mariño (2006). "Improving statistical MT by coupling reordering and decoding". In: *Machine Translation* 20 (3). 10.1007/s10590-007-9024-z, pp. 199–215. ISSN: 0922-6567. URL: http://dx.doi.org/10.1007/s10590-007-9024-z.

DeNero, John, Dan Gillick, James Zhang, and Dan Klein (06/2006). "Why Generative Phrase Models Underperform Surface Heuristics". In: *Proceedings on the Workshop on Statistical Machine Translation*. New York City: Association for Computational Linguistics, pp. 31–38. URL: http://www.aclweb.org/anthology/W/W06/W06-3105.

Foster, George, Roland Kuhn, and Howard Johnson (07/2006). "Phrasetable Smoothing for Statistical Machine Translation". In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia: Association for Computational Linguistics, pp. 53–61. URL: http://www.aclweb.org/anthology/W/W06/W06-1607.

Galley, Michel, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer (2006). "Scalable Inference and Training of Context-Rich Syntactic Translation Models". In: *Proc. of the 21st ICCL and 44th ACL*. Sydney, Australia, pp. 961–968.

Gao, Sheng, Wen Wu, Chin-Hui Lee, and Tat-Seng Chua (04/2006). "A maximal figure-of-merit (MFoM)-learning approach to robust classifier design for text categorization". In: *ACM Trans. Inf. Syst.* 24.2, pp. 190–218. ISSN: 1046-8188. DOI: 10.1145/1148020.1148022. URL: http://doi.acm.org/10.1145/1148020.1148022.

Habash, Nizar and Fatiha Sadat (2006). "Arabic preprocessing schemes for statistical machine translation". In: *NAACL-HLT*. New York, New York: Association for Computational Linguistics, pp. 49–52.

Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman (2006). *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. ISBN: 0321455363.

Ittycheriah, Abe, Yasser Al-Onaizan, and Salim Roukos (2006). *The IBM Arabic-English Word Alignment Corpus*. Tech. rep.

Krishnan, V. and C. D. Manning (2006). "An effective two-stage model for exploiting non-local dependencies in named entity recognition". In: *ICCL and ACL*. Sydney, Australia: Association for Computational Linguistics, pp. 1121–1128. DOI: http://dx.doi.org/10.3115/1220175.1220316.

Kumar, Shankar, Yonggang Deng, and William Byrne (03/2006). "A weighted finite state transducer translation template model for statistical machine translation". In: *Nat. Lang. Eng.* 12 (1), pp. 35–75. ISSN: 1351-3249.

Lacoste-Julien, Simon, Ben Taskar, Dan Klein, and Michael I. Jordan (06/2006). "Word Alignment via Quadratic Assignment". In: *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. New York City, USA: Association for Computational Linguistics, pp. 112–119. URL: http://www.aclweb.org/anthology/N/N06/N06-1015.

Liang, Percy, Ben Taskar, and Dan Klein (06/2006). "Alignment by Agreement". In: *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. New York City, USA: Association for Computational Linguistics, pp. 104–111. URL: http://www.aclweb.org/anthology/N/N06/N06-1014.

Liang, Percy, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar (2006). "An end-to-end discriminative approach to machine translation". In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Lin-*

guistics. ACL-44. Sydney, Australia: Association for Computational Linguistics, pp. 761–768. DOI: http://dx.doi.org/10.3115/1220175.1220271. URL: http://dx.doi.org/10.3115/1220175.1220271.

Lopez, Adam and Philip Resnik (08/2006). "Word-Based Alignment, Phrase-Based Translation: What's the Link?" In: *5th Conference of the Association for Machine Translation in the Americas (AMTA)*. Boston, Massachusetts.

Mariño, José B., Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José A. R. Fonollosa, and Marta Ruiz Costa-jussà (2006). "N-gram-based Machine Translation". In: *Computational Linguistics* 32.4.

Moore, Robert C., Wen-tau Yih, and Andreas Bode (07/2006). "Improved Discriminative Bilingual Word Alignment". In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, pp. 513–520. URL: http://www.aclweb.org/anthology/P/P06/P06-1065.

Plas, Lonneke van der and Jörg Tiedemann (2006). "Finding synonyms using automatic word alignment and measures of distributional similarity". In: *Proceedings of the COLING/ACL on Main conference poster sessions*. COLING-ACL '06. Sydney, Australia: Association for Computational Linguistics, pp. 866–873. URL: http://dl.acm.org/citation.cfm?id=1273073.1273184.

Snover, Matthew, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul (08/2006a). "A Study of Translation Edit Rate with Targeted Human Annotation". In: *5th Conference of the Association for Machine Translation in the Americas (AMTA)*. Boston, Massachusetts.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul (2006b). "A study of translation edit rate with targeted human annotation". In: *In Proceedings of Association for Machine Translation in the Americas*, pp. 223–231.

Suzuki, Jun, Erik McDermott, and Hideki Isozaki (2006). "Training conditional random fields with multivariate evaluation measures". In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. ACL-44. Sydney, Australia: Association for Computational Linguistics, pp. 217–224. DOI: 10.3115/1220175.1220203. URL: http://dx.doi.org/10.3115/1220175.1220203.

Udupa, Raghavendra and Hemanta Kumar Maji (2006). "Computational Complexity of Statistical Machine Translation". In: *EACL*.

Vilar, David, Maja Popovic, and Hermann Ney (11/2006). "AER: do we need to "improve" our alignments?" In: *Proc. of the International Workshop on Spoken Language Translation*. Kyoto, Japan.

Villada Moirón, Begoña and Jörg Tiedemann (04/2006). "Identifying idiomatic expressions using automatic word-alignment". In: *Proceedings of the EACL 2006 Workshop on Multiword Expressions in a Multilingual Context*. Trento, Italy. URL: http://www.aclweb.org/anthology/W/W06/.

Andrew, Galen and Jianfeng Gao (2007). "Scalable Training of L1-Regularized Log-Linear Models". In: *ICML*, pp. 33–40.

Blanchon, Hervé and Christian Boitet (2007). "Pour l'évaluation des systèmes de TA par des méthodes externes fondées sur la tâche". In: *TAL* 48.1, pp. 33–65.

Cherry, Colin and Dekang Lin (04/2007). "Inversion Transduction Grammar for Joint Phrasal Translation Modeling". In: *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*. Rochester, New York: Association for Computational Linguistics, pp. 17–24. URL: http://www.aclweb.org/anthology/W/W07/W07-0403.

Davis, Paul, Zhuli Xie, and Kevin Small (2007). "All Links are not the Same: Evaluating Word Alignments for Statistical Machine Translation". In: *Proceedings of the MT Summit XI*.

DeNero, John and Dan Klein (06/2007). "Tailoring Word Alignments to Syntactic Machine Translation". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 17–24. URL: http://www.aclweb.org/anthology/P/P07/P07-1003.

Deng, Yonggang and Yuqing Gao (06/2007). "Guiding Statistical Word Alignment Models With Prior Knowledge". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 1–8. URL: http://www.aclweb.org/anthology/P/P07/P07-1001.

Deng, Yonggang, Shankar Kumar, and William Byrne (2007). "Segmentation and alignment of parallel text for statistical machine translation". In: *Natural Language Engineering* 13.3, pp. 235–260.

Dreyer, Markus, Keith Hall, and Sanjeev Khudanpur (2007). "Comparing reordering constraints for SMT using efficient Bleu oracle computation". In: *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*. SSST '07. Rochester, New York: Association for Computational Linguistics, pp. 103–110. URL: http://dl.acm.org/citation.cfm?id=1626281.1626295.

Elming, Jakob and Nizar Habash (04/2007). "Combination of Statistical Word Alignments Based on Multiple Preprocessing Schemes". In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. Rochester, New York: Association for Computational Linguistics, pp. 25–28. URL: http://www.aclweb.org/anthology/N/N07/N07-2007.

Euzenat, Jérôme and Pavel Shvaiko (2007). *Ontology matching*. Heidelberg (DE): Springer-Verlag. ISBN: 3-540-49611-4.

Foster, George and Roland Kuhn (06/2007). "Mixture-Model Adaptation for SMT". In: *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, pp. 128–135. URL: http://www.aclweb.org/anthology/W/W07/W07-0217.

Fraser, Alexander (2007). "Improved word alignments for statistical machine translation". AAI3291804. PhD thesis. Los Angeles, CA, USA. ISBN: 978-0-549-39015-2.

Fraser, Alexander and Daniel Marcu (2007a). "Getting the Structure Right for Word Alignment: LEAF". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 51–60. URL: http://www.aclweb.org/anthology/D/D07/D07-1006.

Fraser, Alexander and Daniel Marcu (2007b). "Measuring Word Alignment Quality for Statistical Machine Translation". In: *Computational Linguistics* 33.3.

Graça, João, Kuzman Ganchev, and Ben Taskar (2007). "Expectation Maximization and Posterior Constraints". In: *NIPS*.

Habash, Nizar (2007). "Arabic Morphological Representations for Machine Translation". In: *Arabic Computational Morphology*. Ed. by Abdelhadi Soudi, Antal van den Bosch, Günter Neumann, and Nancy Ide. Vol. 38. Text, Speech and Language Technology. Springer Netherlands, pp. 263–285. ISBN: 978-1-4020-6046-5.

Johnson, Howard, Joel Martin, George Foster, and Roland Kuhn (2007). "Improving Translation Quality by Discarding Most of the Phrasetable". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 967–975. URL: http://www.aclweb.org/anthology/D/D07/D07-1103.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher J. Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst (06/2007). "Moses: Open Source Toolkit for Statistical Machine Translation". In: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, pp. 177–180. URL: http://www.aclweb.org/anthology/P/P07/P07-2045.

Lambert, Patrik, Rafael E. Banchs, and Josep M. Crego (04/2007). "Discriminative Alignment Training without Annotated Data for Machine Translation". In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. Rochester, New York: Association for Computational Linguistics, pp. 85–88. URL: http://www.aclweb.org/anthology/N/N07/N07-2022.

May, Jonathan and Kevin Knight (2007). "Syntactic Re-Alignment Models for Machine Translation". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 360–368. URL: http://www.aclweb.org/anthology/D/D07/D07-1038.

Moore, Robert C. and Chris Quirk (2007). "An iteratively-trained segmentation-free phrase translation model for statistical machine translation". In: *Proceedings of the Second Workshop on Statistical Machine Translation*. StatMT '07. Prague, Czech Republic: Association for Computational Linguistics, pp. 112–119. URL: http:

//dl.acm.org/citation.cfm?id=1626355.1626370.

Nguyen, Patrick, Milind Mahajan, and Xiaodong He (06/2007). "Training Non-Parametric Features for Statistical Machine Translation". In: *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, pp. 72–79. URL: http://www.aclweb.org/anthology/W/W07/W07-0210.

Ren, Dengjun, Hua Wu, and Haifeng Wang (2007). "Improving Statistical Word Alignment with Various Clues". In: *Proceedings of the MT Summit XI*.

Riezler, Stefan, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu (06/2007). "Statistical Machine Translation for Query Expansion in Answer Retrieval". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 464–471. URL: http://www.aclweb.org/anthology/P07-1059.

Sutton, Charles and Andrew Mccallum (2007). "An Introduction to Conditional Random Fields for Relational Learning". In: *Introduction to Statistical Relational Learning*. Ed. by Lise Getoor and Ben Taskar. MIT Press. URL: http://www.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf.

Sutton, Charles, Andrew McCallum, and Khashayar Rohanimanesh (03/2007). "Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data". In: *Journal of Machine Learning Research* 8, pp. 693–723.

Venkatapathy, Sriram and Aravind Joshi (04/2007). "Discriminative word alignment by learning the alignment structure and syntactic divergence between a language pair". In: *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*. Rochester, New York: Association for Computational Linguistics, pp. 49–56. URL: http://www.aclweb.org/anthology/W/W07/W07-0407.

Zettlemoyer, Luke S. and Robert C. Moore (2007). "Selective phrase pair extraction for improved statistical machine translation". In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers on XX*. NAACL '07. Rochester, New York: Association for Computational Linguistics, pp. 209–212.

Agarwal, Abhaya and Alon Lavie (06/2008). "Meteor, M-BLEU and M-TER: Evaluation Metrics for High-Correlation with Human Rankings of Machine Translation Output". In: *Proceedings of the Third Workshop on Statistical Machine Translation*. Columbus, Ohio: Association for Computational Linguistics, pp. 115–118. URL: http://www.aclweb.org/anthology/W/W08/W08-0312.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder (2008). "Further meta-evaluation of machine translation". In: *Proceedings of the Third Workshop on Statistical Machine Translation*. StatMT '08. Columbus, Ohio: Association for Computational Linguistics, pp. 70–106. ISBN: 978-1-932432-09-1. URL: http://dl.acm.org/citation.cfm?id=1626394.1626403.

Cer, Daniel, Daniel Jurafsky, and Christopher D. Manning (06/2008). "Regularization and Search for Minimum Error Rate Training". In: *Proceedings of the Third Workshop on Statistical Machine Translation*. Columbus, Ohio: Association for Computational Linguistics, pp. 26–34. URL: http://www.aclweb.org/anthology/W/W08/W08-0304.

Chiang, David, Yuval Marton, and Philip Resnik (2008). "Online Large-Margin Training of Syntactic and Structural Translation Features". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 224–233. URL: http://www.aclweb.org/anthology/D08-1024.

Crego, Josep M. and Nizar Habash (06/2008). "Using Shallow Syntax Information to Improve Word Alignment and Reordering for SMT". In: *Proceedings of the Third Workshop on Statistical Machine Translation*. Columbus, Ohio: Association for Computational Linguistics, pp. 53–61. URL: http://www.aclweb.org/anthology/W/W08/W08-0307.

DeNero, John, Alexandre Bouchard-Côté, and Dan Klein (2008). "Sampling alignment structure under a Bayesian translation model". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '08. Honolulu, Hawaii: Association for Computational Linguistics, pp. 314–323. URL: http://dl.acm.org/citation.cfm?id=1613715.1613758.

DeNero, John and Dan Klein (06/2008). "The Complexity of Phrase Alignment Problems". In:

Proceedings of ACL-08: HLT, Short Papers. Columbus, Ohio: Association for Computational Linguistics, pp. 25–28. URL: http://www.aclweb.org/anthology/P/P08/P08-2007.

Deng, Yonggang and William J. Byrne (2008). "HMM Word and Phrase Alignment for Statistical Machine Translation". In: *IEEE Transactions on Audio, Speech & Language Processing* 16.3, pp. 494–507.

Deng, Yonggang, Jia Xu, and Yuqing Gao (06/2008). "Phrase Table Training for Precision and Recall: What Makes a Good Phrase and a Good Phrase Pair?" In: *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, pp. 81–88. URL: http://www.aclweb.org/anthology/P/P08/P08-1010.

Fossum, Victoria Li, Kevin Knight, and Steven Abney (06/2008). "Using Syntax to Improve Word Alignment Precision for Syntax-Based Machine Translation". In: *Proceedings of the Third Workshop on Statistical Machine Translation*. Columbus, Ohio: Association for Computational Linguistics, pp. 44–52. URL: http://www.aclweb.org/anthology/W/W08/W08-0306.

Galley, Michel and Christopher D. Manning (10/2008). "A Simple and Effective Hierarchical Phrase Reordering Model". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 847–855. URL: http://www.aclweb.org/anthology/D08-1089.

Ganchev, Kuzman, Joao V. Graca, and Ben Taskar (06/2008). "Better Alignments = Better Translations?" In: *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, pp. 986–993. URL: http://www.aclweb.org/anthology/P/P08/P08-1112.

Gao, Qin and Stephan Vogel (2008). "Parallel implementations of word alignment tool". In: *SETQA-NLP '08*. Columbus, Ohio, pp. 49–57.

Giménez, Jesús and Lluís Màrquez (06/2008). "A Smorgasbord of Features for Automatic MT Evaluation". In: *Proceedings of the Third Workshop on Statistical Machine Translation*. Columbus, Ohio: Association for Computational Linguistics, pp. 195–198. URL: http://www.aclweb.org/anthology/W/W08/W08-0332.

Jurafsky, Daniel and James H. Martin (2008). *Speech and Language Processing (2nd edition)*. Prentice Hall.

Lardilleux, Adrien and Yves Lepage (08/2008). "Multilingual Alignments by Monolingual String Differences". In: *Coling 2008: Companion volume: Posters and Demonstrations*. Manchester, UK: Coling 2008 Organizing Committee, pp. 53–56. URL: http://www.aclweb.org/anthology/C08-3014.

Lopez, Adam (2008a). "Statistical Machine Translation". In: *ACM Computing Surveys* 40.3.

Lopez, Adam (08/2008b). "Tera-Scale Translation Models via Pattern Matching". In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, pp. 505–512. URL: http://www.aclweb.org/anthology/C08-1064.

Ma, Yanjun, Sylwia Ozdowska, Yanli Sun, and Andy Way (06/2008). "Improving Word Alignment Using Syntactic Dependencies". In: *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*. Columbus, Ohio: Association for Computational Linguistics, pp. 69–77. URL: http://www.aclweb.org/anthology/W/W08/W08-0409.

Martins, A. F. T., D. Das, N. A. Smith, and E. P. Xing (2008). "Stacking dependency parsers". In: *EMNLP*. Honolulu, Hawaii: Association for Computational Linguistics, pp. 157–166.

Moore, Robert C. and Chris Quirk (08/2008). "Random Restarts in Minimum Error Rate Training for Statistical Machine Translation". In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, pp. 585–592. URL: http://www.aclweb.org/anthology/C08-1074.

Niehues, Jan and Stephan Vogel (06/2008). "Discriminative Word Alignment via Alignment Matrix Modeling". In: *Proceedings of the Third Workshop on Statistical Machine Translation*. Columbus, Ohio: Association for Computational Linguistics, pp. 18–25. URL: http://www.aclweb.org/anthology/W/W08/W08-0303.

Roth, Ryan, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin (06/2008). "Arabic Morphological Tagging, Diacritization, and Lemmatization Using Lexeme Models and Feature Ranking". In: *Proceedings of ACL-08: HLT, Short Papers*. Columbus, Ohio: Association for Computational Linguistics, pp. 117–120. URL: http://www.aclweb.org/anthology/P/P08/P08-2030.

Sagot, Benoît and Darja Fišer (2008). "Building a free French wordnet from multilingual resources". In: *OntoLex*. Marrakech, Morocco.

Sharma, K.R. (2008). *Bioinformatics: Sequence Alignment and Markov Models*. McGraw-Hill. ISBN: 9780071593069. URL: http://books.google.fr/books?id=3mcYYJaXXx0C.

Venugopal, Ashish, Andreas Zollmann, Noah A. Smith, and Stephan Vogel (2008). "Wider Pipelines: N-Best Alignments and Parses in MT Training". In: *Proceedings of the Association for Machine Translation in the Americas (AMTA)*.

Zhang, Bangzuo and Wanli Zuo (2008). "Learning from Positive and Unlabeled Examples: A Survey". In: *Proceedings of the 2008 International Symposiums on Information Processing*. Washington, DC, USA: IEEE Computer Society, pp. 650–654. ISBN: 978-0-7695-3151-9. DOI: 10.1109/ISIP.2008.79. URL: http://portal.acm.org/citation.cfm?id=1437902.1438917.

Zhang, Hao, Chris Quirk, Robert C. Moore, and Daniel Gildea (06/2008). "Bayesian Learning of Non-Compositional Phrases with Synchronous Parsing". In: *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, pp. 97–105. URL: http://www.aclweb.org/anthology/P/P08/P08-1012.

Andrés-Ferrer, Jesús and Alfons Juan (05/2009). "A phrase-based hidden semi-Markov approach to machine translation." In: *Procedings of European Association for Machine Translation (EAMT)*. Barcelona, Spain: European Association for Machine Translation.

Blunsom, Phil, Trevor Cohn, Chris Dyer, and Miles Osborne (2009). "A Gibbs sampler for phrasal synchronous grammar induction". In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*. ACL '09. Suntec, Singapore: Association for Computational Linguistics, pp. 782–790. ISBN: 978-1-932432-46-6. URL: http://dl.acm.org/citation.cfm?id=1690219.1690256.

Bourdaillet, Julien, Stéphane Huet, Fabrizio Gotti, Guy Lapalme, and Philippe Langlais (2009). "Enhancing the Bilingual Concordancer TransSearch with Word-level Alignment". In: *Canadian AI 2009*. Lecture Notes in Artifical Intelligence. Kelowna, BC, Canada.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, and Josh Schroeder (2009). "Findings of the 2009 workshop on statistical machine translation". In: *Proceedings of the Fourth Workshop on Statistical Machine Translation*. StatMT '09. Athens, Greece: Association for Computational Linguistics, pp. 1–28. URL: http://dl.acm.org/citation.cfm?id=1626431.1626433.

Chiang, David, Kevin Knight, and Wei Wang (2009). "11,001 New features for statistical machine translation". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. NAACL '09. Boulder, Colorado: Association for Computational Linguistics, pp. 218–226. ISBN: 978-1-932432-41-1.

Crego, Josep Maria and François Yvon (2009). "Gappy translation units under left-to-right SMT decoding". In: *Proceedings of the meeting of the European Association for Machine Translation (EAMT)*. Barcelona, Spain, pp. 66–73. URL: http://www.mt-archive.info/EAMT-2009-Crego.pdf.

Deng, Yonggang and Bowen Zhou (08/2009). "Optimizing Word Alignment Combination For Phrase Table Training". In: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Suntec, Singapore: Association for Computational Linguistics, pp. 229–232. URL: http://www.aclweb.org/anthology/P/P09/P09-2058.

Dyer, Chris (2009). "Using a maximum entropy model to build segmentation lattices for MT". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. NAACL '09. Boulder, Colorado: Association for Computational Linguistics, pp. 406–414. ISBN: 978-1-932432-41-1. URL: http://dl.acm.org/citation.cfm?id=1620754.1620814.

Galron, Daniel, Sergio Penkale, Andy Way, and I. Dan Melamed (2009). "Accuracy-based scoring for DOT: towards direct error minimization for data-oriented translation". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*. EMNLP '09. Singapore: Association for Computational Linguistics, pp. 371–380. ISBN: 978-1-932432-59-6.

Guzman, Francisco, Qin Gao, and Stephan Vogel (2009). "Reassessment of the Role of Phrase Extraction". In: *12th MT Summit*.

Haghighi, Aria, John Blitzer, John DeNero, and Dan Klein (2009). "Better word alignments with supervised ITG models". In: *Proceedings of the Joint Conference of the 47th Annual Meet-*

ing of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2. ACL '09. Suntec, Singapore: Association for Computational Linguistics, pp. 923–931. ISBN: 978-1-932432-46-6. URL: http://dl.acm.org/citation.cfm?id=1690219.1690276.

Huang, Fei (08/2009). "Confidence Measure for Word Alignment". In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Computational Linguistics, pp. 932–940. URL: http://www.aclweb.org/anthology/P/P09/P09-1105.

Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Liu, Yang, Tian Xia, Xinyan Xiao, and Qun Liu (2009). "Weighted alignment matrices for statistical machine translation". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*. EMNLP '09. Singapore: Association for Computational Linguistics, pp. 1017–1026.

Madsen, Mathias Winther (2009). "The Limits of Machine Translation". MA thesis. Copenhagen: Departement of Scandinavian Studies and Linguistics, Faculty of Humanities, University of Copenhagen.

N. Habash, O. Rambow and R. Roth (04/2009). "MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization". In: *Proc. of the Second International Conf. on Arabic Language Resources and Tools*. Ed. by Khalid Choukri and Bente Maegaard. Cairo, Egypt: The MEDAR Consortium. ISBN: 2-9517408-5-9.

Russell, S. J. and P. Norvig (2009). *Artificial Intelligence: A Modern Approach*. 3rd. Prentice Hall.

Saers, Markus and Dekai Wu (2009). "Improving phrase-based translation via word alignments from stochastic inversion transduction grammars". In: *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*. SSST '09. Boulder, Colorado: Association for Computational Linguistics, pp. 28–36. ISBN: 978-1-932432-39-8. URL: http://dl.acm.org/citation.cfm?id=1626344.1626348.

Søgaard, Anders (2009). "On the complexity of alignment problems in two synchronous grammar formalisms". In: *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*. SSST '09. Boulder, Colorado: Association for Computational Linguistics, pp. 60–68. ISBN: 978-1-932432-39-8. URL: http://dl.acm.org/citation.cfm?id=1626344.1626352.

Søgaard, Anders and Jonas Kuhn (2009). "Empirical lower bounds on alignment error rates in syntax-based machine translation". In: *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*. SSST '09. Boulder, Colorado: Association for Computational Linguistics, pp. 19–27. ISBN: 978-1-932432-39-8. URL: http://dl.acm.org/citation.cfm?id=1626344.1626347.

Søgaard, Anders and Dekai Wu (2009). "Empirical lower bounds on translation unit error rate for the full class of inversion transduction grammars". In: *Proceedings of the 11th International Conference on Parsing Technologies*. IWPT '09. Paris, France: Association for Computational Linguistics, pp. 33–36. URL: http://dl.acm.org/citation.cfm?id=1697236.1697243.

Tomeh, Nadi, Nicola Cancedda, and Marc Dymetman (08/2009). "Complexity-based Phrase-table Filtering for Statistical Machine Translation". In: *MT Summit XII: proceedings of the twelfth Machine Translation Summit*. Ottawa, Ontario, Canada, pp. 144–151.

Tsuruoka, Y., J. Tsujii, and S. Ananiadou (2009). "Stochastic Gradient Descent Training for L1-regularized Log-linear Models with Cumulative Penalty". In: *ACL-IJCNLP 2009*, pp. 477–485. URL: http://www.aclweb.org/anthology/P/P09/P09-1054.pdf.

Zaidan, Omar F. (2009). "Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems". In: *The Prague Bulletin of Mathematical Linguistics* 91, pp. 79–88.

Allauzen, A. and G. Wisniewski (2010). "Modèles discriminants pour l'alignement mot-à-mot". In: *Traitement Automatique des Langues (TAL)* 50.3/2009.

Berg-Kirkpatrick, Taylor, Alexandre Bouchard-Côté, John DeNero, and Dan Klein (06/2010). "Painless Unsupervised Learning with Features". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 582–590. URL: http://www.aclweb.org/anthology/N10-1083.

DeNero, John and Dan Klein (2010). "Discriminative modeling of extraction sets for machine translation". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Uppsala, Sweden: Association for Computational Linguistics, pp. 1453–1463.

Galley, Michel and Christopher D. Manning (06/2010). "Accurate Non-Hierarchical Phrase-Based Translation". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 966–974.

Gimpel, Kevin and Noah A. Smith (2010). "Softmax-margin CRFs: training log-linear models with cost functions". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10. Los Angeles, California: Association for Computational Linguistics, pp. 733–736. ISBN: 1-932432-65-5. URL: http://dl.acm.org/citation.cfm?id=1857999.1858111.

Gispert, Adrià de, Juan Pino, and William Byrne (2010). "Hierarchical phrase-based translation grammars extracted from alignment posterior probabilities". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. EMNLP '10. Cambridge, Massachusetts: Association for Computational Linguistics, pp. 545–554.

Graça, João, Kuzman Ganchev, and Ben Taskar (2010). "Learning tractable word alignment models with complex constraints". In: *Comput. Linguist.* 36 (3), pp. 481–504. ISSN: 0891-2017.

Habash, Nizar (2010). *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Hovelynck, Matthijs and Boris Chidlovskii (2010). "Multi-modality in one-class classification". In: *Proceedings of the 19th international conference on World wide web*. WWW '10. Raleigh, North Carolina, USA: ACM, pp. 441–450. ISBN: 978-1-60558-799-8. DOI: http://doi.acm.org/10.1145/1772690.1772736. URL: http://doi.acm.org/10.1145/1772690.1772736.

Koehn, Philipp (2010). *Statistical Machine Translation*. 1st. New York, NY, USA: Cambridge University Press. ISBN: 0521874157, 9780521874151.

Lardilleux, Adrien, Julien Gosme, and Yves Lepage (2010). "Bilingual Lexicon Induction: Effortless Evaluation of Word Alignment Tools and Production of Resources for Improbable Language Pairs". In: *LREC*.

Lavergne, Thomas, Olivier Cappé, and François Yvon (07/2010). "Practical Very Large Scale CRFs". In: *ACL*.

Ling, Wang, Tiago Luís, Joao Graça, Luísa Coheur, and Isabel Trancoso (2010). "Towards a General and Extensible Phrase-Extraction Algorithm". In: *Proc. of 7th IWSLT*. Paris, France, pp. 313–320.

Liu, Shujie, Chi-Ho Li, and Ming Zhou (2010). "Discriminative Pruning for Discriminative ITG Alignment." In: *ACL*. Ed. by Jan Hajic, Sandra Carberry, and Stephen Clark. The Association for Computer Linguistics, pp. 316–324.

Pauls, Adam, Dan Klein, David Chiang, and Kevin Knight (2010). "Unsupervised syntactic alignment with inversion transduction grammars". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10. Los Angeles, California: Association for Computational Linguistics, pp. 118–126. ISBN: 1-932432-65-5. URL: http://dl.acm.org/citation.cfm?id=1857999.1858013.

Penkale, Sergio, Yanjun Ma, Daniel Galron, and Andy Way (2010). "Accuracy-Based Scoring for Phrase-Based Statistical Machine Translation". In: *AMTA 2010: The Ninth Conference of the Association for Machine Translation in the Americas*. Denver, CO., pp. 257–266.

Riesa, Jason and Daniel Marcu (2010). "Hierarchical search for word alignment". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Uppsala, Sweden: Association for Computational Linguistics, pp. 157–166. URL: http://dl.acm.org/citation.cfm?id=1858681.1858698.

Saers, Markus, Joakim Nivre, and Dekai Wu (2010). "Word alignment with Stochastic Bracketing Linear Inversion Transduction Grammar". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10. Los Angeles, California: Association for Computational Linguistics, pp. 341–344. ISBN: 1-932432-65-5. URL: http://dl.acm.org/citation.cfm?id=1857999.1858049.

Tomeh, Nadi, Alexandre Allauzen, Guillaume Wisniewski, and François Yvon (2010). "Refining

Word Alignment with Discriminative Training". In: *Proceedings of the ninth Conference of the Association for Machine Translation in the America (AMTA)*. Denver, CO.

Wu, Dekai (2010). "Alignment". In: *Handbook of Natural Language Processing, Second Edition*. Ed. by Nitin Indurkhya and Fred J. Damerau. ISBN 978-1420085921. Boca Raton, FL: CRC Press, Taylor and Francis Group.

Bansal, Mohit, Chris Quirk, and Robert C. Moore (2011). "Gappy phrasal alignment by agreement". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. HLT '11. Portland, Oregon: Association for Computational Linguistics, pp. 1308–1317. ISBN: 978-1-932432-87-9. URL: http://dl.acm.org/citation.cfm?id=2002472.2002635.

DeNero, John and Klaus Macherey (2011). "Model-based aligner combination using dual decomposition". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. HLT '11. Portland, Oregon: Association for Computational Linguistics, pp. 420–429. ISBN: 978-1-932432-87-9. URL: http://dl.acm.org/citation.cfm?id=2002472.2002526.

Dyer, Chris, Jonathan H. Clark, Alon Lavie, and Noah A. Smith (06/2011). "Unsupervised Word Alignment with Arbitrary Features". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 409–419. URL: http://www.aclweb.org/anthology/P11-1042.

Esplà, Miquel, Felipe Sánchez-Martínez, and Mikel L. Forcada (2011). "Using word alignments to assist computer-aided translation users by marking which target-side words to change or keep unedited". In: *Proceedings of the 15th Annual Conference of the European Association for Machine Translation*. Ed. by Mikel L. Forcada, Heidi Depraetere, and Vincent Vandeghinste. European Association for Machine Translation. Leuven, Belgium, pp. 81–89.

Fischer, Andreas, Volkmar Frinken, Alicia Fornés, and Horst Bunke (2011). "Transcription alignment of Latin manuscripts using hidden Markov models". In: *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*. HIP '11. Beijing, China: ACM, pp. 29–36. ISBN: 978-1-4503-0916-5. DOI: 10.1145/2037342.2037348. URL: http://doi.acm.org/10.1145/2037342.2037348.

Saers, Markus and Dekai Wu (2011). "Principled induction of phrasal bilexica". In: *Proceedings of the European Conference on Machine Translation*. Ed. by Mikel Forcada and Heidi Depraetere. Leuven, Belgium, pp. 313–320.

Smith, Noah A. (05/2011). *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Sontag, David, Amir Globerson, and Tommi Jaakkola (2011). "Introduction to Dual Decomposition for Inference". In: *Optimization for Machine Learning*. Ed. by Suvrit Sra, Sebastian Nowozin, and Stephen J. Wright. MIT Press.

Specia, Lucia (05/2011). "Exploiting objective annotations for measuring translation post-editing effort". In: *Proceedings of the 15th conference of the European Association for Machine Translation*. Leuven, Belgium, pp. 73–80.

Tiedemann, Jörg (2011). *Bitext Alignment*. Synthesis Lectures on Human Language Technologies, Graeme Hirst (ed) 14. Morgan & Claypool Publishers.

Tomeh, Nadi, Alexandre Allauzen, and François Yvon (2011a). "Discriminative Weighted Alignment Matrices for Statistical Machine Translation". In: *Proceedings of the European Conference on Machine Translation*. Ed. by Mikel Forcada and Heidi Depraetere. Leuven, Belgium, pp. 305–312.

Tomeh, Nadi, Allexandre Allauzen, and François Yvon (06/2011b). "Estimation d'un modèle de traduction à partir d'alignements mot-à-mot non-déterministes". In: *Proceedings of the 18th TALN Conference (Traitement Automatique des Langues Naturelles), TALN-2011*. Montpellier, France.

Tomeh, Nadi, Thomas Lavergne, Allexandre Allauzen, and François Yvon (2011a). "Designing an Improved Discriminative Word Aligner". In: *Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING)*. Tokyo, Japan.

Tomeh, Nadi, Marco Turchi, Guillaume Wisniewski, Alexandre Allauzen, and François Yvon (2011b). "How Good Are Your Phrases? Assessing Phrase Quality with Single Class Classification". In: *Proceedings of the heigth International Workshop on Spoken Language Translation (IWSLT)*. Ed. by Mei-Yuh Hwang and Sebastian Stüker. San Francisco, CA, pp. 261–268.

Toutanova, Kristina and Michel Galley (2011). "Why initialization matters for IBM model 1: multiple optima and non-strict convexity". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*. HLT '11. Portland, Oregon: Association for Computational Linguistics, pp. 461–466. ISBN: 978-1-932432-88-6. URL: http://dl.acm.org/citation.cfm?id=2002736.2002829.

Turchi, M. and M. Ehrmann (2011). "Knowledge Expansion of a Statistical Machine Translation System using Morphological Resources". In: *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics (CILCING)*, pp. 37–43.

Dreyer, Markus and Daniel Marcu (06/2012). "HyTER: Meaning-Equivalent Semantics for Translation Evaluation". In: *Proceedings of the The 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. NAACL:HLT'12. Montreal, Canada.