

PhD Dissertation

ARCHITECTURE AND MODELING
FOR N -GRAM-BASED
STATISTICAL MACHINE TRANSLATION

Josep M. Crego Clemente

Thesis advisor

Prof. Dr. José B. Mariño Acebal



TALP Research Center, Speech Processing Group
Department of Signal Theory and Communications
Universitat Politècnica de Catalunya

Barcelona, February 2008

Abstract

This Ph.D. thesis dissertation addresses several aspects of Statistical Machine Translation (SMT). The emphasis is put on the architecture and modeling of an SMT system developed during the last few years at the Technical University of Catalunya (UPC).

A detailed study of the different system components is conducted. It is built following the N -gram-based approach to SMT. Mainly, it models the translation process by means of a joint-probability translation model introduced in a log-linear combination of bilingual N -grams with additional feature functions. A comparison is carried out against a standard phrase-based system to allow for a deeper understanding of its main features.

One of the main contributions of this thesis work is the implementation of a search algorithm. It is based on dynamic programming and specially designed to work over N -gram-based translation models. Apart from the underlying translation model, it contrasts to other search algorithms by the introduction of several feature functions under the well known log-linear framework and by allowing for a tight coupling with source-side reorderings.

A source words reordering approach based on linguistic information is proposed. Mainly, it aims at reducing the complexity of the translation process derived of the structural differences (word order) of language pairs. Reordering is presented as the problem of introducing into the input sentence source words the necessary permutations to acquire the word order of the target language. With the objective of reducing the reordering errors, the reordering problem is tightly coupled with the overall search by means of decoding a permutation graph which contains the best scored reordering hypotheses. The use of different linguistic information (Part-Of-Speech tags, chunks, full parse trees) and techniques to accurately predict reorderings is evaluated. Efficiency and accuracy results are shown over a wide range of data size translation tasks with different reordering needs.

Resum

Aquesta tesi doctoral està dedicada a l'estudi de varis aspectes dels sistemes de traducció automàtica estocàstica (TAE). Molt especialment, a l'estructura i modelat del sistema de TAE desenvolupat durant els darrers anys a la Universitat Politècnica de Catalunya (UPC).

Es realitza un estudi detallat de les diferents components del sistema. El sistema està construït basat en l'enfoc per N -grames bilingües. Aquest enfoc permet estimar un model de traducció de probabilitat conjunta per mitjà de la combinació, dins un entorn log-linial, de cadenes d' N -grames i funcions caraterístiques addicionals. També es presenta una comparativa amb un sistema estàndard basat en sintagmes amb l'objectiu d'aprofundir en la comprensió del sistema estudiat.

Una de les contribucions més importants d'aquesta tesi consisteix en la implementació de l'algoritme de búsqueda. Està construït utilitzant tècniques de programació dinàmica i dissenyat especialment per treballar amb un model de traducció basat en N -grames bilingües. A més de pel model de traducció subjacent, l'algoritme es diferencia d'altres algoritmes de cerca pel fet d'introduir varies funcions caraterístiques dins l'entorn log-linial i pel fort acoplament d'aquest amb els reordenaments de la frase d'entrada.

Es proposa la introducció de reordenaments a la frase d'entrada basats en informació lingüística amb l'objectiu de reduir les diferències estructurals del parell de llengües, reduint així la complexitat del procés de traducció. El procés de reordenament es presenta com el problema de trobar les permutacions de les paraules de la frase d'entrada que fan que aquesta estigui expressada en l'estructura (ordre de paraules) del llenguatge destí. Amb l'objectiu d'evitar els errors produïts en el procés de reordenament, la decisió final de reordenament es realitza a la cerca global, a través de la decodificació d'un graf de permutacions que conté les hipòtesis de reordenament més probables. S'avalua la utilització d'informació lingüística (etiquetes morfo-sintàctiques, chunks, arbres sintàctics) en el procés de reordenament. Els resultats d'eficiència i qualitat es presenten per varies tasques de diferent tamany i necessitats de reordenament.

Agraïments

Voldria donar les gràcies a tots aquells que han fet possible aquesta tesi.

En primer lloc al José Mariño, qui sense cap mena de dubtes ha estat el director de tesi que qualsevol doctorand voldria tenir, tant en l'aspecte docent i científic com humà.

També vull donar les gràcies als companys del grup de traducció estocàstica de la UPC, amb els quals treballar ha estat en tot moment un plaer. El Patrik, la Marta, el Rafa, el Max, l' Adrián, i molt especialment a l'Adrià, amb qui em sento en deute des del primer moment per l'enorme ajuda prestada. En gran mesura aquesta també és la seva tesi.

Gràcies també al conjunt de companys amb els quals m'ha tocat compartir molts moments al llarg de més de quatre anys. Entre d'altres el Jordi, el Pablo, la Marta, el Pere, el Jan, la Mireia, la Mònica, el Frank, el Cristian, l'Enric, *etc.*.

També vull agrair l'excel·lent acolliment rebut al Center for Computational Learning Systems de la Columbia University durant els mesos d'estada a la ciutat de New York. Molt especialment al Nizar.

Per acabar, vull donar les gràcies als meus pares, a la meva germana i a la Marie. Sense els quals, per infinits motius, aquesta tesi no hauria estat mai possible.

Moltes gràcies a tots,

Josep Maria

Barcelona, Desembre de 2007

Contents

1	Introduction	1
1.1	Machine Translation	1
1.1.1	Brief History	2
1.1.2	Current Approaches	2
1.1.3	Statistical Machine Translation	4
1.2	Scientific Goals	5
1.3	Thesis Organization	6
1.4	Research Contributions	7
2	State of the art	9
2.1	Noisy Channel	9
2.1.1	Word Alignment	10
2.1.2	Phrase-based Translation Models	12
2.2	Log-linear Feature Combination	13
2.2.1	Minimum Error Training	14
2.2.2	Re-scoring	14
2.3	Search in SMT	15
2.3.1	Evolution	15
2.3.2	Reordering	16
2.3.3	Search as Parsing	19
2.4	Machine Translation Evaluation	20
2.4.1	Automatic Metrics	21
2.4.2	Human Metrics	28
3	<i>N</i>-gram-based approach to Statistical Machine Translation	31

3.1	Introduction	32
3.2	Bilingual N -gram Translation Model	32
3.2.1	From Word-alignments to Translation Units	32
3.2.2	N -gram Language Model Estimation	38
3.3	N -gram-based SMT System	40
3.3.1	Log-linear Combination of Feature Functions	40
3.3.2	Training Scheme	41
3.3.3	Optimization Work	43
3.4	Experiments	43
3.4.1	Tuple Extraction and Pruning	45
3.4.2	Translation and Language N -gram Size	47
3.4.3	Source-NULled Tuple Strategy Comparison	48
3.4.4	Feature Function Contributions	49
3.4.5	Error Analysis	50
3.5	Contrasting Phrase-based SMT	52
3.5.1	Phrase-based Translation Model	52
3.5.2	Translation Accuracy Under Different Data Size Conditions	55
3.6	Chapter Summary and Conclusions	56
4	Linguistically-motivated Reordering Framework	59
4.1	Introduction	60
4.1.1	Related Work	60
4.1.2	N -gram-based Approach to SMT	61
4.2	Reordering Framework	62
4.2.1	Unfold Tuples / Reordering Rules	63
4.2.2	Input Graph Extension	74
4.2.3	Distortion Modeling	77
4.3	Experiments	78
4.3.1	Common Details	78
4.3.2	Spanish-English Translation Task	79
4.3.3	Arabic-English Translation Task	84
4.3.4	Chinese-English Translation Task	89

4.4	Chapter Summary and Conclusions	92
5	Decoding Algorithm for N-gram-based Translation Models	95
5.1	Introduction	96
5.1.1	Related Work	96
5.1.2	N -gram-based Approach to SMT	96
5.2	Search Algorithm	97
5.2.1	Permutation Graph	98
5.2.2	Core Algorithm	99
5.2.3	Output Graph	103
5.2.4	Contrasting Phrase-based Decoders	103
5.2.5	Speeding Up the Search	107
5.3	Additional Feature Functions	110
5.3.1	Additional Translation Models	111
5.3.2	Target N -gram Language Model	111
5.3.3	Word/Tuple Bonus	111
5.3.4	Reordering Model	112
5.3.5	Tagged-target N -gram Language Model	112
5.3.6	Tagged-source N -gram Language Model	113
5.4	Chapter Summary and Conclusions	115
6	Conclusions and Future Work	117
6.1	Future Work	118
A	Corpora Description	119
A.1	EPPS Spanish-English	119
A.1.1	EPPS Spanish-English ver1	120
A.1.2	EPPS Spanish-English ver2	120
A.1.3	EPPS Spanish-English ver3	121
A.2	NIST Arabic-English	121
A.3	BTEC Chinese-English	122
B	Participation in MT Evaluations	123
B.1	TC-Star 3rd Evaluation	124

B.2 IWSLT 2007	126
B.3 ACL 2007 WMT	128
B.4 NIST 2006 MT Evaluation	130
C Publications by the author	133
Bibliography	137

List of Figures

1.1	<i>Machine Translation pyramid</i>	3
1.2	<i>Architecture of a SMT system.</i>	5
2.1	<i>Illustration of the generative process underlying IBM models</i>	11
2.2	<i>Phrase extraction from a certain word aligned pair of sentences.</i>	13
2.3	<i>Multiple stacks used in a beam-based search.</i>	16
2.4	<i>Permutations graph of a monotonic (top) and reordered (bottom) search.</i>	17
2.5	<i>Word order harmonization strategy.</i>	18
2.6	<i>NIST penalty graphical representation</i>	23
3.1	<i>Three tuple segmentations of the sentence pair: 'Maria finalmente abofeteó a la bruja # Maria finally slapped the witch'.</i>	33
3.2	<i>Tuple extraction from a word-to-word aligned pair of sentences.</i>	34
3.3	<i>Tuple extraction from a certain word aligned pair of sentences.</i>	35
3.4	<i>Estimation of a 'bilingual' N-gram language model using the SRILM toolkit.</i> . . .	39
3.5	<i>Feature estimation of an N-gram-based SMT system from parallel data. Flow diagram.</i>	42
3.6	<i>Optimization procedure. Flow diagram.</i>	44
3.7	<i>Phrase and tuple extraction.</i>	53
3.8	<i>Phrase and tuple extraction with noisy alignments.</i>	53
3.9	<i>Generative process. Phrase-based (left) and N-gram-based (right) approaches.</i> . .	54
4.1	<i>Tuples (top right) extracted from a given word aligned sentence pair (top left) and permutation graph (bottom) of the input sentence: 'how long does the trip last today'.</i>	62
4.2	<i>Generative translation process when introducing the reordering framework.</i>	63
4.3	<i>Pattern extraction.</i>	64

4.4	<i>Tuple extraction following the unfold technique.</i>	65
4.5	<i>1-to-N alignments can not be unfold (left). Envisaged solution (right).</i>	66
4.6	<i>Tuples (top right) extracted from a given word aligned sentence pair (top left) after 'unfolding' the source words and permutation graph (bottom) of the input sentence: 'how long does the trip last today'.</i>	67
4.7	<i>Linguistic information used in reordering rules.</i>	68
4.8	<i>POS-based and chunk-based Rule extraction.</i>	71
4.9	<i>Constituency (up) and dependency (down) parsing trees.</i>	72
4.10	<i>Extraction of syntax-based reordering rules. Chinese words are shown in simplified Chinese.</i>	73
4.11	<i>Extraction of syntax-based reordering rules. Rule generalization.</i>	74
4.12	<i>Input graph extension.</i>	75
4.13	<i>Two rules are used to extend the reordering graph of a given input sentence.</i>	76
4.14	<i>Source POS-tagged N-gram language model.</i>	78
4.15	<i>In Spanish the order of the Subject, Verb and Object are interchangeable.</i>	80
4.16	<i>Wrong pattern extraction because of erroneous word-to-word alignments.</i>	81
4.17	<i>An example of long distance reordering of Arabic VSO order into English SVO order</i>	84
4.18	<i>Refinement of word alignments using chunks.</i>	86
4.19	<i>Linguistic information, reordering graph and translation composition of an Arabic sentence.</i>	89
4.20	<i>Two Chinese sentences with identical words and different meaning ('LE' is an aspect particle indicating completion/change).</i>	90
4.21	<i>Nouns and modifiers in Chinese ('DE' precedes a noun and follows a nominal modifier.</i>	90
5.1	<i>Generative process introducing distortion. Phrase-based (left) and N-gram-based (right) approaches.</i>	97
5.2	<i>Reordering graph (up) and confusion network (down) formed for the 1-best input sentence 'ideas excelentes y constructivas'.</i>	99
5.3	<i>Monotonic input graph and its associated search graph for an input sentence with J input words.</i>	101
5.4	<i>Reordered input graph and its associated search graph for the input sentence 'ideas excelentes y constructivas'.</i>	101
5.5	<i>Fields used to represent a hypothesis.</i>	102
5.6	<i>Different translations (%) in the N-best list.</i>	104

5.7	<i>Oracle results (WER) regarding the size of the N-best list.</i>	105
5.8	<i>Phrase-based and N-gram-based search errors.</i>	106
5.9	<i>Phrase-based and N-gram-based search graphs.</i>	106
5.10	<i>Reordering input graph created using local constraints ($l = 3$).</i>	108
5.11	<i>Efficiency results under different reordering conditions.</i>	109
5.12	<i>Extended set of fields used to represent a hypothesis.</i>	113
5.13	<i>Memory access derived of an N-gram call.</i>	114

List of Tables

3.1	<i>Model size and translation accuracy derived of the alignment set used to extract translation units.</i>	46
3.2	<i>Model size and translation accuracy derived of the tuple vocabulary pruning. . . .</i>	46
3.3	<i>Perplexity measurements for translation and target language models of different N-gram size</i>	47
3.4	<i>Evaluation results for experiments on N-gram size incidence.</i>	48
3.5	<i>Evaluation results for experiments on strategies for handling source-NULLED tuples.</i>	49
3.6	<i>Evaluation results for experiments on feature function contribution.</i>	50
3.7	<i>Percentage of occurrence for each type of error in English-to-Spanish and Spanish-to-English translations that were studied</i>	51
3.8	<i>Models used by each system.</i>	55
3.9	<i>Accuracy results under different training data size conditions.</i>	56
4.1	<i>Spanish-to-English (top) and English-to-Spanish (bottom) reordering rules.</i>	80
4.2	<i>Evaluation results for experiments with different translation units, N-gram size and additional models. Spanish-to-English translation task.</i>	81
4.3	<i>Evaluation results for experiments with different translation units, N-gram size and additional models. English-to-Spanish translation task.</i>	82
4.4	<i>Evaluation results for experiments on the impact of the maximum size of the POS-based rules. Spanish-to-English translation task.</i>	83
4.5	<i>Evaluation results for experiments on the impact of the maximum size of the POS-based rules. English-to-Spanish translation task.</i>	83
4.6	<i>Reorderings hypothesized for the test set according to their size.</i>	83
4.7	<i>Arabic, Spanish and English Linguistic Features</i>	85
4.8	<i>Evaluation results for experiments on translation units and N-gram size incidence. Arabic-English translation task.</i>	87
4.9	<i>Reorderings hypothesized and employed in the 1-best translation output according to their size. BLEU scores are shown for each test set.</i>	88

4.10	<i>Evaluation results for experiments on translation units and N-gram size incidence. Chinese-English translation task.</i>	91
4.11	<i>Reorderings hypothesized and employed in the 1-best translation output according to their size. BLEU scores are shown for each test set.</i>	92
5.1	<i>Histogram pruning (beam size).</i>	110
5.2	<i>Threshold pruning.</i>	110
5.3	<i>Caching technique results.</i>	115
A.1	<i>EPPS ver1. Basic statistics for the training, development and test data sets . . .</i>	120
A.2	<i>EPPS ver2. Basic statistics for the training, development and test data sets. . .</i>	120
A.3	<i>EPPS ver3. Basic statistics for the training, development and test data sets. . .</i>	121
A.4	<i>NIST Arabic-English corpus. Basic statistics for the training (train), development (MT02) and test data sets (MT03, MT04, MT05).</i>	121
A.5	<i>BTEC Chinese-English corpus. Basic statistics for the training (train), development (dev1) and test data sets (dev2, dev3).</i>	122
B.1	<i>TC-Star'07 Spanish-English automatic (BLEU/NIST) comparative results for the three tasks (FTE, Verbatim and ASR) and corpus domains (Europarl and Cortes). Site Rank is shown in parentheses.</i>	125
B.2	<i>TC-Star'07 English-Spanish automatic (BLEU/NIST) comparative results for the three tasks (FTE, Verbatim and ASR). Site Rank is shown in parentheses for each measure. Europarl task.</i>	126
B.3	<i>IWSLT'07 Arabic-English human (%Better) and automatic (BLEU) comparative results for the two tasks (Clean and ASR). Site Rank is shown in parentheses for each measure.</i>	127
B.4	<i>IWSLT'07 Chinese-English human (%Better) and automatic (BLEU) comparative results for the Clean task. Site Rank is shown in parentheses for each measure.</i>	128
B.5	<i>WMT'07 Spanish-English human (Adequacy/Fluency) and automatic (METEOR/BLEU) comparative results for the two tasks (Europarl and News). Site Rank is shown in parentheses for each measure.</i>	129
B.6	<i>WMT'07 English-Spanish human (Adequacy/Fluency) and automatic (METEOR/BLEU) comparative results for the two tasks (Europarl and News). Site Rank is shown in parentheses for each measure.</i>	130
B.7	<i>NIST'06 Arabic-English and Chinese-English comparative results (in terms of BLEU) for the two subsets (NIST and GALE) of the large data condition. . . .</i>	131

Chapter 1

Introduction

Without doubt, the globalized society we live in has a growing demand for immediate and accurate information. Nowadays, it is technologically easy to provide an economically cheap and fast access to this information to the majority of the population. However, language remains an important barrier that prevents all information from being spread across different cultures because of the high cost in terms of money and time that human translation implies.

Among others, and without aiming at being exhaustive, demands for translation can be found on communities with several official languages (such as Canada, Switzerland, Spain, the European Union, *etc.*), companies with interests spread all over the world, or in general, a generalized wish of humans to fully understand the vast amount of information that everyday is made available all around the world.

In special, the popularity of Internet provides an interesting mechanism from which to collect extremely large amounts of multilingual information. Despite that most of this information is released without the corresponding translation, the every-day growing availability of human-translated examples (parallel corpora) as well as the enormous improvement in performance of current computers has made rise the optimism among scientists in the MT community.

Specially from the mid nineties, the statistical machine translation (SMT) approach (based on the use of large amounts of parallel corpora to estimate statistical models describing the translation process) has gained in popularity in contrast to previous approaches (based on linguistic knowledge representations). A reason for this success is found on the relatively easy development of systems with enough competence as to achieve rather competitive results.

1.1 Machine Translation

In this thesis, we understand machine translation (MT) as the process that takes a message (from its textual representation) in a source language and transforms it into a target language, keeping the exact meaning. Hence, words and their underlying structure are supposed to change while meaning must remain unchanged.

1.1.1 Brief History

The beginnings of machine translation (SMT) can be traced back to the early fifties, closely related to the ideas from which information theory arose [Sha49b] and inspired by works on cryptography [Sha49a, Sha51] during World War II. According to this view, machine translation was conceived as the problem of finding a sentence by decoding a given “encrypted” version of it [Wea55].

Several research projects were devoted to MT during the fifties. However, the complexity of the linguistic phenomena involved, together with the computational limitations of the time did not allow to reach high quality automatic translations, what made the initial enthusiasm disappear, at the time that funding and research.

As a matter of example of the generalized depression feeling of that time, the Bar-Hillel report [BH60] concluded that Fully Automatic High-Quality Translation was an unreachable goal and that research efforts should be focused on less-ambitious tasks, such as Computer-assisted Machine Translations tools.

During the 1970s, research on MT was retaken thanks in part to the growing demands for translations in multilingual societies (such as Canada and Europe). Many research projects have led MT to be established as a research field and as a commercial application [Arn95].

Since initially documented, MT has revealed to be one of the most complex tasks to carry out in the field of natural language processing (NLP), considered one of the AI-hard problems.

1.1.2 Current Approaches

We will next review the major research approaches in machine translation. Several criteria can be used to distinguish MT systems. The most popular considers the level of linguistic analysis (and generation) required by the MT system. This can be graphically expressed by the machine translation pyramid in Figure 1.1.

Typically, three different types of MT systems are distinguished: *direct* approach, *transfer* approach and *interlingua* approach.

- The simplest approach, represented by the bottom of the pyramid, is the *direct* approach. Systems within this approach do not perform any kind of linguistic analysis of the source sentence in order to produce a target sentence. Translation is performed on a word-by-word basis. The approach was basically followed by the early MT systems. Nowadays, this preliminary approach has been abandoned, even in the framework of corpus-based approaches (see below).
- In the *transfer* approach, the translation process is decomposed into three steps: analysis, transfer and generation. The source sentence is analyzed producing an abstract representation. In the transfer step the representation is transferred into a corresponding representation in the target language. Finally, the generation step produces the target sentence from this intermediate representation. Usually, rules to achieve the three steps are collected manually, thus involving a great amount of expert human effort. Apart from that, when several competing rules can be applied, it is difficult for the systems to prioritize them, as there is no natural way to weigh them. This approach was massively followed in the

1980s, and despite much research effort, high-quality MT was only achieved for limited domains [Hut92].

- Finally, the *interlingua* approach produces a deep syntactic and semantic analysis of the source sentence (language independent interlingua representation), turning the translation task into generating a target sentence according to the obtained interlingua representation. This approach advocates for the deepest analysis of the source sentence. The interlingua language, has the advantage that, once the source meaning is captured by it, we can express it in any number of target languages, so long as a generation engine for each of them exists. Several drawbacks make this approach unpractical from a conceptual point of view. On the one hand, the difficulty of creating the interlingua conceptual language. Which must be capable of bearing the particular semantics of all languages. Additionally, the requirement that the whole source sentence needs to be understood before being translated, has proved to make the approach less robust to the ungrammatical expressions of informal language, typically produced by automatic speech recognition systems.

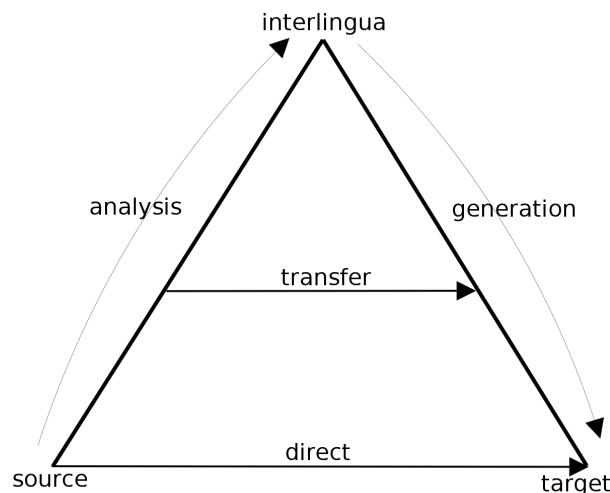


Figure 1.1: *Machine Translation pyramid*

MT systems can also be classified according to the core technology they use. Under this classification we find *rule-based* and *corpus-based* approaches.

- In the *rule-based approach*, human experts specify a set of rules, aiming at describing the translation process. This approach conveys an enormous work of human experts [Hut92, Dor94, Arn95].
- Under the *corpus-based approach*, the knowledge is automatically extracted by analyzing translation examples from a parallel corpus (built by human experts). The advantage is that, once the required techniques have been developed for a given language pair, (in theory) MT systems can be very quickly developed for new language pairs provided training data. A corpus-based approach typically follows a direct or transfer approach.

Within the corpus-based approaches we can further distinguish between *example-based* MT and *statistical* MT.

- *Example-based* MT (EBMT) makes use of previously seen examples in parallel corpora. A translation is provided by choosing and combining these examples in an appropriate way.
- In *Statistical* MT (SMT), parallel examples are used to train a statistical translation model. Thus, relying on statistical parameters and a set of translation and language models, among other data-driven features. This approach initially worked on a word-by-word basis (hence classified as a direct method). However, current systems attempt to introduce a certain degree of linguistic analysis into the SMT approach, slightly climbing up the aforementioned MT pyramid.

The following section further introduces the statistical approach to MT.

1.1.3 Statistical Machine Translation

The SMT approach was introduced more than a decade ago when IBM researchers presented the *Candide* SMT system [Bro90, Bro93]. The approach has seen an increasing interest because of different factors, which range from the growing availability of parallel data, together with the increasing computational performance, to the successful results achieved in several evaluation campaigns ¹, which are proved to be as good (or even better) than results of system following the rule-based approach.

SMT can be seen as a decision problem where among the whole sentences in a target language, it has to be found the most likely to be the translation of a given source sentence. The likelihood of a target sentence to be the translation of a source sentence is learnt from a bilingual text corpus.

This probability is typically learnt for small segments (sequences of words). Thus, building translation as a composition of partial translations.

As far as the set of sentences in a target language is infinite, a subset is only taken into account. Generally, the considered subset is structured in partial translation hypotheses that are to be composed in a search process. In the first SMT systems, these partial hypotheses were composed of single words (one source and one target word), therefore considering words to be the translation units of the process. Later, this units were expanded to include several words (in both source and target sides).

Figure 1.2 illustrates the basic architecture of a SMT system. It is divided into two main steps: training, where the system is built from available translation examples; and test, where new sentences are being translated.

The first training process consists of a word-to-word alignment automatically induced from the parallel corpus (previously aligned in a sentence-to-sentence basis). Further tokenization/categorization processes of the training corpus can also be considered as part of the SMT architecture, prior to the word alignment.

Partial translation units are automatically extracted from the training parallel corpus, according to the word alignments previously extracted.

¹See NIST annual evaluation results at <http://www.nist.gov/speech/tests/mt>

Translation units are used when decoding new sentences (test). In the search, several models are typically used to account for the adequacy/fluency of translation options.

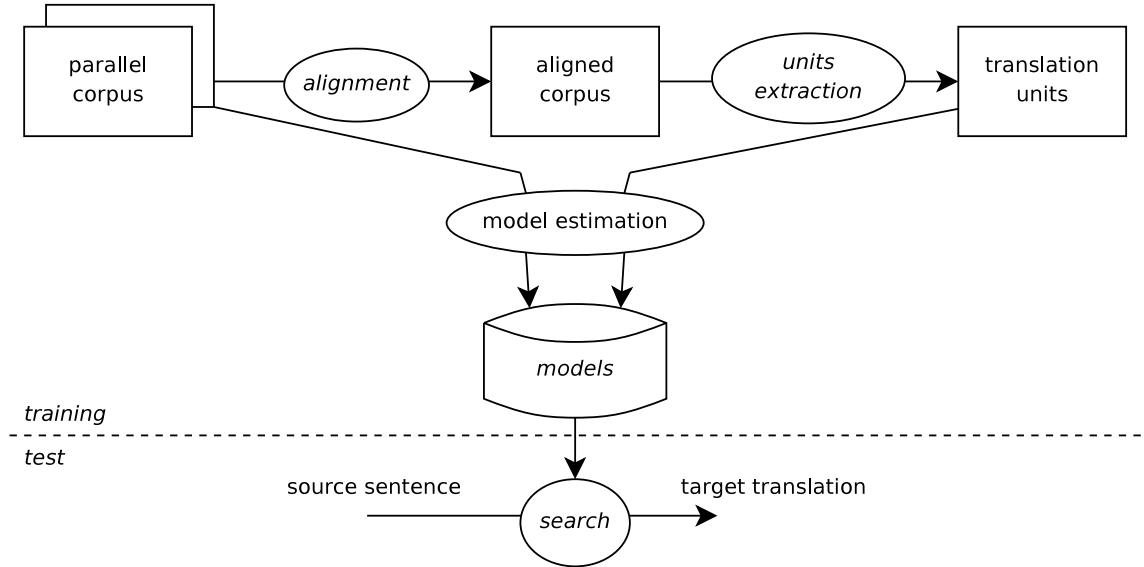


Figure 1.2: *Architecture of a SMT system.*

The SMT approach is more formally introduced in §2.

1.2 Scientific Goals

The aim of this work is to extend the state-of-the-art in SMT. The main objectives pursued by this Ph.D. thesis consist of the following:

To further redefine the architecture and modeling of the system. In order to attain state-of-the-art results for a system, research must be continuously carried out. From these research efforts (jointly addressed by all UPC SMT researchers) our system has continuously been upgraded with new capabilities. Among others we can cite the implementation of several feature functions, an optimization tool, a re-scoring tool, the use of reordering in the search, *etc.* Thanks to the many changes introduced, our SMT system has grown achieving comparable results to other outstanding systems.

To study and introduce a linguistically-motivated reordering framework.

Throughout this research work, we have been interested in trying to overcome the current limitations of SMT. One of the main limitations consists of the difficulty to deal with language pairs with different word order. When word reordering is taken into account, the complexity of the translation process turns it into an extremely hard problem, which needs for introducing additional information sources and decoding techniques to be handled. We have mainly tackled this problem by introducing a reordering framework with two main features:

- *Use of linguistic information to more accurately predict the target word order*, where different linguistic information has been used in order to account for the systematic differences of the language pairs and to achieve enough generalization power as to predict unseen examples. Information sources range from Part-Of-Speech tags to syntax parse trees.
- *Tightly couple the reordering decision to the global search*, by means of a permutation graph that encodes a restricted set of reorderings which are to be decoded in the search. Thus, the final decision is taken more informed in the global search, where the whole information sources (models) are available.

To develop a decoding tool for N -gram-based translation models. At the time when this research work began, the UPC Speech Processing Group, with long-standing experience on Automatic Speech Recognition (ASR), had initiated research on the SMT field only two years ago (2001). Thus, lacking of many software tools but with the clear idea of developing the concept of a joint-probability translation model, initially implemented with a Finite-State Transducer (FST). Hence, the purpose of developing a search algorithm that should allow using larger data sets as well as introducing additional information sources in the system set the foundations of this Ph.D. research work. The search algorithm has been under the focus all along the duration of this Ph.D. as it is a key component of the SMT system. Mainly, any technique aiming at dealing with a translation problem needs for a decoder extension to be implemented and carefully coupled.

1.3 Thesis Organization

This Ph.D. thesis dissertation is divided into **six** chapters. This introductory chapter is followed by an overview on the various statistical machine translation approaches that have been and are being applied in the field, with an emphasis on related works on decoding and reordering. The next three chapters are devoted to the presentation of the thesis contributions. The final chapter concludes and outlines further work.

Outline of the thesis dissertation:

Chapter 2 presents an overview of Statistical Machine Translation. It starts with the mathematical foundations of SMT, that can be traced back to the early nineties with the apparition of word-based translation models. Next, we detail the introduction of phrase-based translation models and a mathematical framework where multiple models can be introduced log-linearly combined. Special attention is payed to the search algorithms proposed and the introduction of word reordering.

Chapter 3 is dedicated to a detailed study of the N -gram-based approach to SMT. Firstly, it is introduced the particular translation model, which is based on bilingual N -grams. Accurate details are given of the extraction and refinement of translation units. The system incorporates additional models under the well-known maximum entropy framework. Empirical results are reported together with a manual error analysis which emphasizes the strong and weak points of the system. At the end of the chapter, the system is compared to a standard phrase-based system to further accentuate the particularities of each.

Chapter 4 extends the system detailed in the previous chapter with new techniques and models developed to account for word reordering. We have followed a linguistically-informed word monotonization approach to tackle the divergences in word order of the source and target languages. Instead of performing a hard reordering decision in preprocessing, we introduce a tight coupling between reordering and decoding by means of a permutation graph that encodes the most promising reordering hypotheses at a very low computational cost. Translation units are extracted according with the reordering approach, enabling the use of the N -gram translation model as reordering model. Several linguistic information sources are employed and evaluated for the task of learning/generalizing valid reorderings from the training data.

Chapter 5 analyzes the singularities of the search algorithm that works as decoding tool of the N -gram-based translation system. A deep study is carried out from an algorithmic point of view. Efficiency results are given to complement the accuracy results provided in previous chapters. The decoder mainly features a beam search, based on dynamic programming, extended with reordering abilities by means of an input permutations graph. A caching technique is detailed at the end of the chapter which provides further efficiency results.

Chapter 6 draws the main conclusions from this Ph.D. thesis dissertation and details future lines of research extending the work carried out.

At the end of the document the reader can find three appendices. The first **Appendix A** gives details of the corpora used all along this work. Then, **Appendix B** details some of the participations of the UPC N -gram-based system in several international translation evaluations. Finally, **Appendix C** reports a list of the publications by the author related to the Ph.D. work.

1.4 Research Contributions

The main contributions of this Ph.D. thesis dissertation are here summarized:

- *Description and evolution of an N -gram-based SMT system.* Many extensions have been incorporated into the system from the initial bilingual N -gram translation model implementation to the currently state-of-the-art system. We discuss and empirically evaluate different design decisions. Several translation tasks are used to assess the adequacy of the approach proposed. Notice that description and evolution of the system has been a joint research task carried out with other researchers at the technical university of Catalunya (UPC).
- *Introduction of word reordering into the N -gram-based SMT system.* The initial description of the system makes it difficult the introduction of word reordering. The use of an N -gram-based translation model as main feature, mainly estimated by relying on the sequence of bilingual units, complicates the apparition of distortion in the model. However, we have introduced a level of distortion in the extraction process of translation units that not only enables the use of reordering but also allows using the N -gram translation model as reordering model. However, the use of bilingual raw words (by the N -gram translation model) gives very poor generalization power in the task of learning reusable reorderings. Hence, new information sources are introduced which mitigate this problem.

- *Implementation of an N -gram-based SMT decoder.* In parallel with (or as part of) the evolution of the N -gram-based approach to SMT, we have developed a search algorithm that as main feature incorporates an N -gram translation model. It shares many characteristics with standard phrase-based decoders but also introduces new ones which aim at improving accuracy results (in terms of translation and search). Apart from the underlying N -gram translation model it features the ability to traverse a permutation graph (encoding the set of promising reorderings) and the introduction of several models in the log-linear combination of feature functions it implements.

The findings presented in this Ph.D. dissertation work were published in a number of publications, which will be referred to in their respective sections and summarized at the end of the document in Appendix C.

Chapter 2

State of the art

This chapter introduces in the form of an overview the most relevant issues in statistical machine translation.

Firstly, §2.1 outlines the mathematical foundations of SMT introduced by IBM researchers in the early nineties. In that time, the translation process was thought in a word by word basis. This initial section introduces also the notions of word alignment and the evolution from word-based to phrase-based translation models which no longer consider single words as their translation units.

Afterwards, §2.2 introduces the maximum entropy approach leading to the prevailing log-linear combination of feature functions (models). It provides a robust framework which makes it easy the use of additional information sources in the translation process. The framework is responsible of achieving current state-of-the-art results. Details of the system optimization and re-ranking (re-scoring) work are also given in this section.

In §2.3 we outline the most important contributions in SMT decoding. Different decoding algorithms have been used since the beginnings of SMT which implement the overall search that SMT is founded on. Next, the word reordering problem is discussed. It has introduced a level of complexity in current SMT systems that makes the search unfeasible when allowing for unrestricted reorderings. Several alternatives to constraint the search have appeared aiming at alleviating the search problem.

To conclude the chapter, §2.4 provides a detailed overview of the most important automatic evaluation measures, which are widely used by the MT community as well as all along this research work.

2.1 Noisy Channel

Statistical machine translation is based on the assumption that every sentence t in a target language is a possible translation of a given sentence s in a source language. The main difference between two possible translations of a given sentence is a probability assigned to each, which is to be learned from a bilingual text corpus. The first SMT models applied these probabilities to words, therefore considering words to be the translation units of the process.

Supposing we want to translate a source sentence s into a target sentence t , we can follow a noisy-channel approach (regarding the translation process as a channel which distorts the target sentence and outputs the source sentence) as introduced in [Bro90], defining statistical machine translation as the optimization problem expressed by:

$$\hat{t} = \arg \max_{t \in \tau} Pr(t | s) \quad (2.1)$$

where τ is the set of all possible sentences in a target language.

Typically, Bayes rule is applied, obtaining the following expression:

$$\hat{t} = \arg \max_{t \in \tau} Pr(s | t) \cdot Pr(t) \quad (2.2)$$

This way, translating s becomes the problem of detecting which t , among all possible sentences in a target language τ , scores best given the product of two models: $Pr(t)$, the *target language model*, and $Pr(s | t)$, the *translation model*.

The use of such a target language model justifies the application of Bayes rule, as this model helps penalizing non-grammatical target sentences during the search.

2.1.1 Word Alignment

Whereas the language model, typically implemented using N -grams, was already being successfully used in speech processing and other fields, the translation model was first presented by introducing a hidden variable a to account for the alignment relationships between words in each language, as in equation 2.3.

$$Pr(s | t) = \sum_a Pr(s, a | t) = Pr(J | t) \prod_{j=1}^J Pr(a_j | s_1^{j-1}, a_1^{j-1}, t) \cdot Pr(s_j | s_1^{j-1}, a_1^j, e) \quad (2.3)$$

where s_j stands for word in position j of the source sentence s , J is the length of this sentence (in number of words), and a_j stands for the alignment of word s_j , *i.e.* the position in the target sentence t where the word which aligns to s_j is placed.

The set of model parameters, or probabilities, is to be automatically learnt from parallel data. In order to train this huge amount of parameters, in [Bro93] the EM algorithm with increasingly complex models is used. These models are widely known as the five IBM models, and are inspired by the generative process described in Figure 2.1, which interprets the model decomposition of equation 2.3.

Conceptually, this process states that for each target word, we first find how many source words will be generated (following a model denoted as *fertility*); then, we find which source words are generated from each target word (lexicon or *word translation* probabilities); and finally, we reorder the source words (according to a *distortion* model) to obtain the source sentence¹.

¹Note that the process generates the source language from the target, due to the application of Bayes rule in equation 2.2.

The alignment models introduced in the previous lines are more formally expressed by:

- $n(\phi|t)$ or Fertility model, which accounts for the probability that a target word t_i generates ϕ_i words in the source sentence.
- $t(s|t)$ or Lexicon model, representing the probability to produce a source word s_j given a target word t_i
- $d(\pi|\tau, \phi, t)$ or Distortion model, which models the probability of placing a source word in position j given that the target word is placed in position i in the target sentence (also used with inverted dependencies, and known as Alignment model)

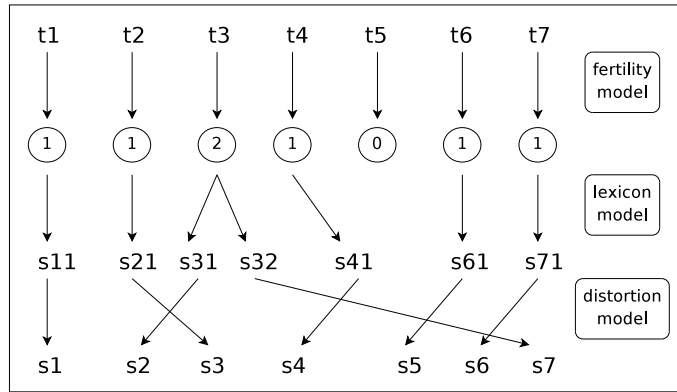


Figure 2.1: Illustration of the generative process underlying IBM models

IBM models 1 and 2 do not include fertility parameters so that the likelihood distributions are guaranteed to achieve a global maximum. Their difference is that Model 1 assigns a uniform distribution to alignment probabilities, whereas Model 2 introduces a zero-order dependency with the position in the source. [Vog96] presented a modification of Model 2 that introduced first-order dependencies in alignment probabilities, the so-called HMM alignment model, with successful results. Model 3 introduces fertility and Model 4 and 5 introduce more detailed dependencies in the alignment model to allow for jumps, so that all of them must be numerically approximated and not even a local maximum can be guaranteed.

A detailed description of IBM models and their estimation from a parallel corpus can be found in [Bro93]. In [Kni99] an informal yet clarifying tutorial on IBM models can be found.

As explicitly introduced by IBM formulation as a model parameter, word alignment becomes a function from source positions j to target positions i , so that $a(j) = i$. This definition implies that resultant alignment solutions will never contain many-to-many links, but only many-to-one², as only one function result is possible for a given source position j .

Although this limitation does not account for many real-life alignment relationships, in principle IBM models can solve this by estimating the probability of generating the source empty word, which can translate into non-empty target words.

²By many-to-many links those relationships between more than one word in each language are referred, whereas many-to-one links associate more than one source word with a single target word. One-to-one links are defined analogously.

In 1999, the John Hopkins University summer workshop research team on SMT released GIZA (as part of the EGYPT toolkit), a tool implementing IBM models training from parallel corpora and best-alignment Viterbi search, as reported in [AO99], where a decoder for model 3 is also described. This was a breakthrough that enabled many other teams to join SMT research easily. In 2001 and 2003 improved versions of this tool were released, and named GIZA++ [Och03c].

However, many current SMT systems do not use IBM model parameters in their training schemes, but only the *most probable* alignment (using a Viterbi search) given the estimated IBM models (typically by means of GIZA++). Therefore, in order to obtain many-to-many word alignments, usually alignments from source-to-target and target-to-source are performed, applying symmetrization strategies. Several symmetrization algorithms have been proposed, being the most widely known the **union**, **intersection** and **refined** [Och00b] of source-to-target and target-to-source alignments, and the **grow-final-diag** [Koe05a] which employs the previous intersection and union alignments.

2.1.2 Phrase-based Translation Models

By the turn of the century it became clear that in many cases specifying translation models at the level of words turned out to be inappropriate, as much local context seemed to be lost during translation. Novel approaches needed to describe their models according to longer units, typically sequences of consecutive words (or *phrases*).

The first approach using longer translation units was presented in [Och99] and named *Alignment Templates*, which are pairs of generalized phrases that allow word classes and include an *internal* word alignment.

An evolution as well as a simplified version of the previous approach is the so-called phrase-based statistical machine translation presented in [Zen02]. Under this framework, word classes are not used (but the actual words from the text instead), and the translation unit loses internal alignment information, turning into so-called *bilingual phrases*. Mathematically, the next equation expresses the idea:

$$Pr(f_1^J | e_1^I) = \alpha(e_1^I) \cdot \sum_B Pr(\tilde{f}_k | \tilde{e}_k) \quad (2.4)$$

where the hidden variable B is the segmentation of the sentence pair in K bilingual phrases $(\tilde{f}_1^K, \tilde{e}_1^K)$, and $\alpha(e_1^I)$ is assuming the same probability for all segmentations.

The phrase translation probabilities are usually estimated, over all bilingual phrases in the corpus, by relative frequency of the target sequence given the source sequence, as in:

$$Pr(\tilde{f}_k | \tilde{e}_k) = \frac{N(\tilde{f}_k, \tilde{e}_k)}{N(\tilde{e}_k)} \quad (2.5)$$

where bilingual phrases are defined as any pair of source and target phrases that have consecutive words and are consistent with the word alignment matrix. According to this criterion, any sequence of consecutive source words and consecutive target words which are aligned to each other and not aligned to any other token in the sentence, become a phrase. This is exemplified

in Figure 2.2, where eight different phrases are extracted and it is worth noting that $AB\#WY$ is *not* extracted, given the definition constraint. For more details on this criterion, see [Och99] or [Zen02].

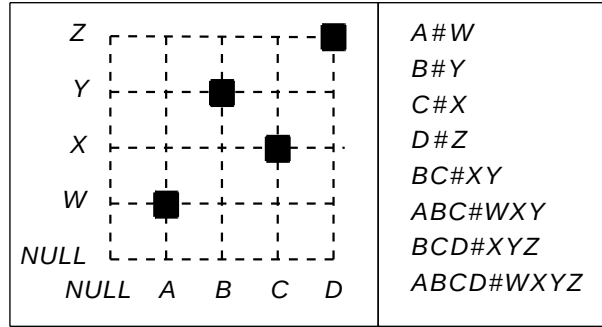


Figure 2.2: *Phrase extraction from a certain word aligned pair of sentences.*

In [Mar02] a joint-probability phrase-based model is introduced, which learns both word and phrase translation and alignment probabilities from a set of parallel sentences. However, this model is only tractable up to an equivalent of IBM model 3, due to severe computational limitations. Furthermore, when comparing this approach to the simple phrase generation from word alignments and a syntax-based phrase generation [Yam01] (discussed in 2.3.3), the approach founded on word alignments achieves the best results, as shown in [Koe03b].

An alternative way to compute phrase translation probabilities is to use IBM model 1 lexical probabilities of the words inside the phrase pair, as presented in [Vog03]. A smoothed relative frequency is used in [Zen04].

Nowadays, many SMT systems follow a phrase-based approach, in that their translation unit is the bilingual phrase, such as [Lee06, Ber06, Mat06, Aru06, Kuh06, Kir06, Hew05], among many others. Most of these systems introduce a log-linear combination of models, as will be discussed in §2.2.

Relevantly, this phrase-based relative frequency model ignores IBM model parameters, being automatically estimated from a word-aligned parallel corpus, thus turning word alignment into a stand-alone training stage which can be done independently.

Lately many tools are being implemented and released, so that every year it becomes easier for a beginner to get quickly introduced into phrase-based SMT, and even run preliminary experiments in one day. Without aiming at completeness, some of them are mentioned here.

Regarding phrase extraction and estimation, an open-source tool has been released in [Ort05].

2.2 Log-linear Feature Combination

An alternative to the noisy-channel approach is to directly model the posterior probability $Pr(t_1^I | s_1^J)$, a well-founded approach in the framework of maximum entropy, as shown in [Ber96]. By treating many different knowledge sources as feature functions, a log-linear combination of models can be performed, allowing an extension of a baseline translation system with the addition

of new feature functions. In this case, the decision rule responds to the following expression:

$$\hat{t}_1^I = \arg \max_{t_1^I \in \tau} \left\{ \sum_{m=1}^M \lambda_m h_m(t_1^I, s_1^J) \right\} \quad (2.6)$$

so that the noisy-channel approach can be obtained as a special case if we consider only two feature functions, namely the target language model $h_1(t_1^I, s_1^J) = \log p(t_1^I)$ and the translation model of the source sentence given the target $h_2(t_1^I, s_1^J) = \log p(s_1^J | t_1^I)$.

2.2.1 Minimum Error Training

This approach, which was introduced in [Pap98] for a natural language understanding task, suggests that the training optimization task becomes finding out the λ_m which weight each model according to a certain criterion. In [Och02] minimum error training is introduced for statistical machine translation, stating that these weights need to be settled by directly minimizing the translation error on a development set, as measured by a certain automatic translation quality measure (see §2.4).

Typically, this log-linear combination includes, apart from a translation model, other feature functions, such as:

- additional language models (word-based or class-based high-order N -grams)
- sentence length models, also called word bonuses
- lexical models (such as IBM model 1 from source to target and from target to source)
- phrase penalties
- others (regarding information on manual lexicon entries or other grammatical features)

In order to optimize the λ_m weights, the usual criterion is to use the maximum posterior probability $p(t|s)$ on a training corpus. Adequate algorithms for such a task are the GIS (Generalized Iterative Scaling) or the *downhill simplex* method [Nel65]. On the other hand, given a loss function based on automatic translation evaluation measures, a minimum bayes-risk decoding scheme can also be used to tune a SMT system, as in [Kum04].

Nowadays, all SMT systems use a log-linear combination of feature models, optimized according to a certain automatic measure on the development data.

2.2.2 Re-scoring

In [She04] a discriminative re-scoring (or re-ranking) strategy is introduced for improving SMT performance (and also used in many systems, such as [Qua05]). This technique works as follows:

- First, a baseline system generates n -best candidate hypotheses

- Then, a set of features which can potentially discriminate between good and bad hypotheses are computed for each candidate
- Finally, these features are weighted in order to produce a new candidate ranking

The advantage is that, given the candidate sentence, features can be computed globally, enabling rapid experimentation with complex feature functions. This approach is followed in [Och03b] and [Och04a] to evaluate the benefits of a huge number of morphological and shallow-syntax feature functions to re-rank candidates from a standard phrase-based system, with little success.

2.3 Search in SMT

As previously stated, SMT is thought as a task where each source sentence s_1^J is transformed into (or generates) a target sentence t_1^I , by means of a stochastic process. Thus, the decoding (search) problem in SMT is expressed by the maximization shown in equations 2.1, 2.2 and 2.6.

2.3.1 Evolution

The first SMT decoders worked at word level: the so-called **word-based** decoders [Bro90], with translation units composed of a single word in the source side. Among these first systems, we find decoders following different search approaches: optimal A* search [Och01], integer programming [Ger01], greedy search algorithms [Ger03] [Ber94] [Wan98]. In [GV03] can be found a detailed study on word-based search algorithms. The difficulty to handle the word order requirements of different languages is a main weakness of these first decoders. In other words, the disparity in word order between languages introduces a level of complexity that is (computationally) very hard to handle by means of word-based decoders, where the problem is approached through permutations of the source words.

Later appeared the **phrase-based** decoders, which use translation candidates composed of arbitrary sequences (without linguistic motivation) of source and target words, commonly called phrases (previously discussed in §2.1.2). The use of phrases allowed to introduce the word context in the translation model which effectively capture short-distance reorderings between language pairs. Thus, alleviating the reordering problem [Til00] [Och04b] [Koe04].

Among the previous decoders, the widely known and successful Pharaoh [Koe04] consists of a freely available beam search phrase-based decoder. Recently, Pharaoh has been replaced/upgraded by Moses [Koe07], which is also a phrase-based decoder implementing a beam search, allowing to input a word lattice and using a factored representation of the raw words (surface forms, lemma, part-of-speech, morphology, word classes, *etc.*). Additionally, a decoder based on confusion networks is presented in [Ber05], and two open-source decoders have been released in [Pat06, Olt06].

Nowadays, many SMT systems employ a phrase-based beam search decoder because of the good performance results it achieves (in terms of accuracy and efficiency). On the one hand, the multiple stacks employed in the search consist of an efficient technique to prune out hypotheses which are fairly compared, allowing high efficiency rates. On the other hand, the use of phrases

as translation units provides the system with a very natural method to give an answer to the problem of modeling reorderings. In special short-distance reorderings, a problem which appears, in different levels, on every language pair.

Figure 2.3 illustrates a beam-based search. The expansion of a given hypothesis (top hypothesis of the second stack) produces new hypotheses which are to be stored in the stack according to the number of target words already translated. Some decoders use the number of source, instead of target, words to select the stack where the new hypotheses are placed.

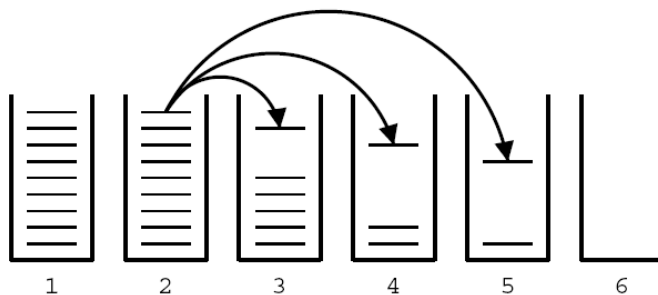


Figure 2.3: *Multiple stacks used in a beam-based search.*

In the last few years a new search strategy has arose motivated by the need to give an answer to long-distance reorderings, for which flat-structured models (such as phrase-based models) fail to give an accurate answer. This new search strategy, founded on the use of parsing technologies, is introduced in §2.3.3. Note that this new approach has radically different structures and parametrization than the aforementioned beam-based search.

Further details on decoding are given on §5, where a freely available N -gram-based SMT decoder is deeply detailed being a major contribution of this thesis work.

2.3.2 Reordering

As previously introduced, reordering is currently one of the major problems in SMT since different languages have different word order requirements. Typically, reordering is introduced in the search by introducing multiple permutations of the input sentence, aiming at acquiring the right word order of the resulting target sentence. However, systems are forced to restrict their distortion abilities because of the high cost in decoding time that permutations imply. In [Kni99], the decoding problem with arbitrary word reorderings is shown to be NP-complete.

Up to date, several alternatives to tackle the reordering problem have been proposed. Despite being a subjective task, we have decided to classify these alternatives into three main groups:

- **Heuristic search constraints**, which do not make use of any linguistic knowledge. They are founded on the application of distance-based restrictions to the search space.
- **Word order monotonization**, where the input sentence word order is transformed in a linguistically-informed preprocessing step in order to harmonize the source and target language word order.

- Use of linguistic information in **re-scoring work**. This alternative has typically provided small accuracy gains given the restriction of being applied to an N -best list.

The previous alternatives are further discussed in the next lines. They all make use of a similar decoder architecture, which needs for minor changes to implement each of them. An additional alternative is also introduced in §2.3.3 where the search is carried out as a parsing process. Hence, a brand new decoder is employed performing a search based on a different architecture and techniques.

2.3.2.1 Heuristic Search Constraints

The first SMT decoders introducing reordering capabilities were founded on the brute force of computers, aiming at finding the best hypothesis through traversing a fully reordered graph (the whole permutations of source-side words are allowed in the search).

This approach is computationally extremely expensive, even for very short input sentences. Therefore, different distance-based constraints were commonly used to make the search feasible: ITG [Wu96], IBM [Ber96], Local [Kan05], MaxJumps [Cre05b], etc. The use of these constraints implies a necessary balance between translation accuracy and efficiency.

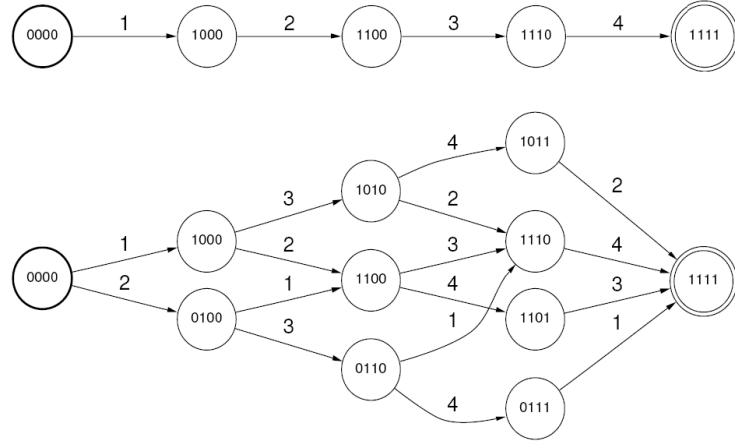


Figure 2.4: *Permutations graph of a monotonic (top) and reordered (bottom) search.*

Figure 2.4 shows the permutations graph computed for a monotonic (top) and a reordered (bottom) search of an input sentence of $J = 4$ words. The reordered graph shows the valid permutations computed following IBM constraints for a value of $l = 2$. IBM constraints allow to deviate from monotonic order by postponing translations up to a limited number of words, *i.e.* at each state, translations can be performed of the first l word positions not yet covered. At each state, the covered words are shown in the form of a bit vector.

In parallel with these heuristic search constraints, a ‘*weak*’ distance-based distortion model was initially used to penalize the longest reorderings, only allowed if sufficiently promoted by the rest of models [Och04b, Koe03b].

Later on, different authors showed that higher accuracy results could be obtained when using

phrase distortion models, allowing for modeling phrase discontinuities. It is the case of the work in [Til04, Koe05a, Kum05], where lexicalized reordering models are proposed. The model learns local orientations (monotonic, non-monotonic) with probabilities for each bilingual phrase from the training material. During decoding, the model attempts to find a Viterbi local orientation sequence. The main problem of this model is the sparseness problem present in the probability estimation.

2.3.2.2 Harmonization of Source and Target Word Order

Similar to the previous heuristic search constraints, the reordering alternative detailed in this section aims at applying a set of permutations to the words of the input sentence to help the system build the translation hypothesis in the right word order.

Word order harmonization was first proposed in [Nie01], where morpho-syntactic information was used to account for the reorderings needed between German and English. In this work reordering was done by prepending German verb prefixes and by treating interrogative sentences using syntactic information. [Xia04] proposes a set of automatically learnt reordering rules (using morpho-syntactic information in the form of POS tags) which are then applied to a French-English translation task. In [Col05a] is used a German parse tree for moving German verbs towards the beginning of the clause. In [Pop06c] POS tag information is used to rewrite the input sentence between Spanish-English and German-English language pairs. [Hab07] employs dependency trees to capture the reordering needs of an Arabic-English translation system.

Figure 2.5 (top) shows how reordering and decoding problems are decoupled under this approach in two main blocks. One of the main drawbacks of this approach is that it takes reordering decisions in a preprocessing step, though, discarding much of the information available in the global search that could play an important role if it was taken into account. So far the reordering problem is only tackled in preprocessing, the errors introduced in this step remain in the final translation output.

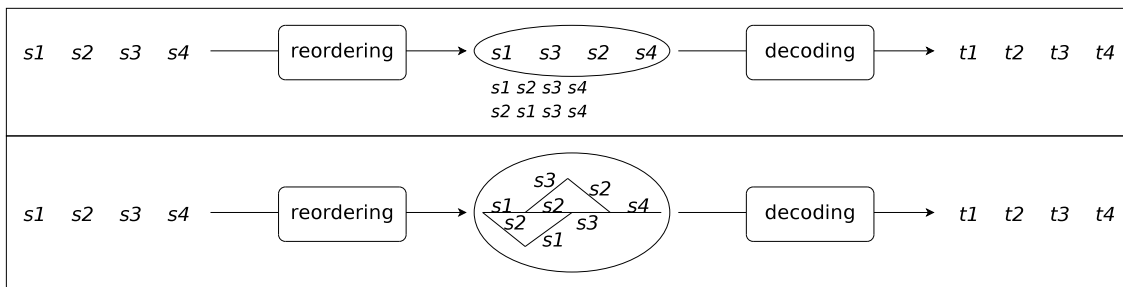


Figure 2.5: *Word order harmonization strategy.*

A natural evolution of the harmonization strategy is shown in Figure 2.5 (bottom), it consists of using a word graph, containing the N -best reordering decisions, instead of the single-best used in the original strategy. The reordering problem is equally approached but alleviating the difficulty of needing high accurate reordering decisions in preprocessing. The final decision is delayed to be taken in the global search (decoding), where all the information is then available.

To the best of our knowledge, reordering graphs were first introduced for SMT in [Zen02],

as a structure used to restrict the number of possible word orders of a fully reordered search. Later, [Cre06a, Cre06b, Cre07b, Zha07] used the same structure to encode linguistically-motivated reorderings. This way re-coupling the decoding and reordering problems by means of a permutation graph which contains linguistically-founded reordering hypotheses. In the previous work, different linguistic information has been used: morphological (Part-Of-Speech tags); shallow syntax (chunks); dependency syntax (parse trees).

Following the same rewriting idea and making use of a permutation graph to couple reordering and decoding, [Cj06] employs a set of automatically learnt word classes instead of linguistic information showing equivalent accuracy results for an Spanish-English task than those shown in [Cre] using POS tag information.

2.3.2.3 Syntactic information in re-scoring work

Re-scoring techniques have also been proposed as a method for using syntactic information to identify translation hypotheses expressed in the right target word order [Koe03a, Och04a, She04]. In these approaches a baseline system is used to generate N -best translation hypotheses. Syntactic features are then used in a second model that re-ranks the N -best lists, in an attempt to improve over the baseline approach. [Koe03a] apply a re-ranking approach to the sub-task of noun-phrase translation.

[Has06] introduces supertag information (or '*almost parsing*' [Ban99]) into a standard phrase-based SMT system in the re-ranking process. It is shown how syntactic constraints can improve translation quality for an Arabic-English translation task. Later, in [Has07] the same researchers introduce the supertag information into the overall search in the form of an additional log-linearly combined model.

2.3.3 Search as Parsing

In spite of the great success of the phrase-based systems, a key limitation of these systems is that they make little or no direct use of syntactic information. However, it appears likely that syntactic information can be of great help in order to accurately modeling many systematic differences [B.94] between the word order of different languages. Ideally, a broad-coverage and linguistically well motivated statistical MT system can be constructed by combining the natural language syntax and machine learning methods.

In recent years, syntax-based statistical machine translation has begun to emerge, aiming at applying statistical models to structured data. Advances in natural language parsing, especially the broad-coverage parsers trained from treebanks, for example [Col99], have made possible the utilization of structural analysis of different languages. The concept of syntax-directed translation was originally proposed in compiling ([E.61, P.68, A.72]), where the source program is parsed into a tree representation that guides the generation of the object code. In other words, the translation is directed by a syntactic tree. In this context, a syntax-directed translator consists of two components, a source language parser and a recursive converter which is usually modeled as a top-down tree-to-string transducer.

A number of researchers ([Als96, Wu97, Yam01, Gil03, Mel04, Gra04, Gal04]) have proposed models where the translation process involves syntactic representations of the source and/or

target languages. One class of approaches make use of '*bitext*' grammars which simultaneously parse both the source and target languages. Another class of approaches make use of syntactic information in the target language alone, effectively transforming the translation problem into a parsing problem. More precisely, Synchronous Tree Adjoining Grammars, proposed by [Shi90], were introduced primarily for semantics but were later also proposed for translation. [Eis03] proposed viewing the MT problem as a probabilistic synchronous tree substitution grammar parsing problem. [Mel03, Mel04] formalized the MT problem as synchronous parsing based on multitext grammars. [Gra04] defined training and decoding algorithms for both generalized tree-to-tree and tree-to-string transducers.

All these approaches, though different in formalism, model the two languages using tree-based transduction rules or a synchronous grammar, possibly probabilistic. The machine translation is done either as a stochastic tree-to-tree transduction or a synchronous parsing process. A further decomposition of these systems can be done by looking at the kind of information they employ. Some of them make use of source and/or target dependency [Qui05, Lan06] or constituent trees, which can be formally syntax-based [Chi05, Wat06] or linguistically syntax-based [Yam02, Wu97, Mar06].

Therefore, syntax-based decoders have emerged aiming at dealing with pair of languages with very different syntactic structures for which the word context introduced in phrase-based decoders is not sufficient to cope with long reorderings. They have gained many adepts because of the significant improvements made by exploiting the power of synchronous rewriting systems.

However, Syntax-directed systems have been typically attacked with the argument of showing a main weakness on their poor efficiency results. However, this argument has been recently overridden by the apparition of new decoders, which show significant improvements when handling with syntactically divergent language pairs under large-scale data translation tasks. An example of such a system can be found in [Mar06], which has obtained state-of-the-art results in Arabic-English and Chinese-English large-sized data tasks.

2.4 Machine Translation Evaluation

Evaluation of Machine Translation has traditionally been performed by humans. While the main criteria that should be taken into account in assessing the quality of MT output are fairly intuitive and well established, the overall task of MT evaluation is both complex and task dependent.

MT evaluation has consequently been an area of significant research over the years. Human evaluation of machine translation output remains the most reliable method to assess translation quality. However, it is a costly and time consuming process.

The development of automatic MT evaluation metrics enables the rapid assessment of systems output. It provides immediate feedback on the effectiveness of techniques applied in the translation process. Additionally, thanks to international evaluation campaigns, these measures have also been used to compare different systems on multiple translation tasks.

2.4.1 Automatic Metrics

As already stated, automatic MT evaluation metrics have made it possible to measure the overall progress of the MT community, as well as reliably compare the success of varying translation systems without relying on expensive and slow human evaluations.

The automatic evaluation of machine translation output is widely accepted as a very difficult task. Typically, the task is performed by producing some kind of similarity/disagreement measure between the translation hypothesis and a set of human reference translations.

The fact that multiple correct alternative translations exist for any input sentence adds complexity to this task. Theoretically, we cannot guarantee that in-correlation with the available set of references means bad translation quality, unless we have all possible correct translations available (which in practice is not possible as it consist of an infinite set).

However, it is accepted that automatic metrics are able to capture progress during system development and to statistically correlate well with human evaluation.

Next, we introduce a set of evaluation metrics which to the best of our knowledge are the most successful in the MT research community (BLEU, NIST, mWER, mPER, METEOR). These metrics also consist of the measures used all along this Ph.D. research work.

2.4.1.1 BLEU score

The BLEU measure (acronym for BiLingual Evaluation Understudy) has dominated most machine translation work. Essentially, it consists of an N -gram corpus-level measure. BLEU was introduced by IBM in [Pap01], and is always referred to a given N -gram order ($BLEU_n$, n usually being 4).

BLEU heavily rewards large N -gram matches between the source and target (reference) translations. Despite being a useful characteristic, this can often unnecessarily penalize syntactically valid but slightly altered translations with low N -gram matches. It is specifically designed to perform the evaluation on a corpus level and can perform badly if used over isolated sentences.

$BLEU_n$ is defined as:

$$BLEU_n = \exp \left(\frac{\sum_{i=1}^n bleu_i}{n} + length_penalty \right) \quad (2.7)$$

where $bleu_i$ and $length_penalty$ are cumulative counts (updated sentence by sentence) referred to the whole evaluation corpus (test and reference sets). Even though these matching counts are computed on a sentence-by-sentence basis, the final score is *not* computed as a cumulative score, *i.e.* it is not computed by accumulating a given sentence score.

Equations 2.8 and 2.9 show $bleu_n$ and $length_penalty$ definitions, respectively:

$$bleu_n = \log \left(\frac{Nmatched_n}{Ntest_n} \right) \quad (2.8)$$

$$length_penalty = \min \left\{ 0, 1 - \frac{shortest_ref_length}{Ntest_1} \right\} \quad (2.9)$$

Finally, $Nmatched_i$, $Ntest_i$ and $shortest_ref_length$ are also cumulative counts (updated sentence by sentence), defined as:

$$Nmatched_i = \sum_{n=1}^N \sum_{ngr \in S} \min \left\{ N(test_n, ngr), \max_r \{ N(ref_{n,r}, ngr) \} \right\} \quad (2.10)$$

where S is the set of N -grams of size i in sentence $test_n$, $N(sent, ngr)$ is the number of occurrences of the N -gram ngr in sentence $sent$, N is the number of sentences to eval, $test_i$ is the i^{th} sentence of the test set, R is the number of different references for each test sentence and $ref_{n,r}$ is the r^{th} reference of the n^{th} test sentence.

$$Ntest_i = \sum_{n=1}^N length(test_n) - i + 1 \quad (2.11)$$

$$shortest_ref_length = \sum_{n=1}^N \min_r \{ length(ref_{n,r}) \} \quad (2.12)$$

From the BLEU description, we can conclude:

- BLEU is a quality metric and it is defined in a range between 0 and 1, 0 meaning the worst-translation (which does not match the references in any word), and 1 the *perfect* translation.
- BLEU is mostly a measure of *precision*, as $bleu_n$ is computed by dividing by the matching n-grams by the number of n-grams in the test (*not* in the reference). In this sense, a very high BLEU could be achieved with a *short* output, so long as all its n-grams are present in a reference.
- The *recall* or *coverage* effect is weighted through the $length_penalty$. However, this is a very rough approach to recall, as it only takes lengths into account.
- Finally, the weight of each effect (precision and recall) might not be clear, being very difficult from a given BLEU score to know whether the provided translation lacks recall, precision or both.

Note that slight variations of these definitions have led to alternative versions of BLEU score, although literature considers BLEU as a unique evaluation measure and no distinction among versions is done. Very recently, an interesting discussion with counterexamples of human correlation was presented in [CB06].

2.4.1.2 NIST score

NIST evaluation metric, introduced in [Dod02], is based on the BLEU metric, but with some alterations. Whereas BLEU simply calculates n-gram precision considering of equal importance each n-gram, NIST calculates how informative a particular n-gram is, and the rarer a correct n-gram is, the more weight it will be given. NIST also differs from BLEU in its calculation of the brevity penalty, and small variations in translation length do not impact the overall score as much.

Again, NIST score is always referred to a given n-gram order ($NIST_n$, usually n being 4), and it is defined as:

$$NIST_n = \left(\sum_{i=1}^n nist_i \right) \cdot nist_penalty \left(\frac{test_1}{\frac{ref_1}{R}} \right) \quad (2.13)$$

where $nist_n$ and $nist_penalty(ratio)$ are cumulative counts (updated sentence by sentence) referred to the whole evaluation corpus (test and reference sets). Even though these matching counts are computed on a sentence-by-sentence basis, the final score is *not* computed as a cumulative score.

The *ratio* value computed using $test_1$, ref_1 and R shows the relation between the number of words of the test set ($test_1$) and the average number of words of the reference sets (ref_1/R). In other words, the relation between the translated number of words and the expected number of words for the whole test set.

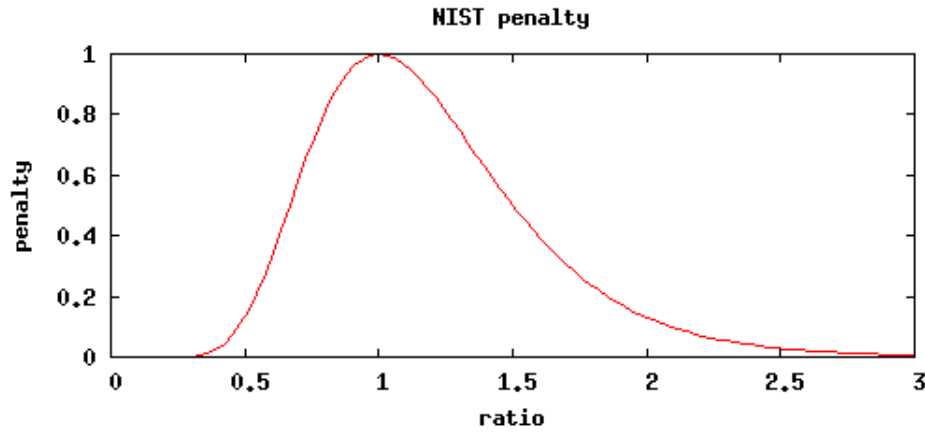


Figure 2.6: NIST penalty graphical representation

Equations 2.14 and 2.15 show $nist_n$ and $nist_penalty$ definitions, respectively. This penalty function is graphically represented in Figure 2.6.

$$nist_n = \frac{Nmatch_weight_n}{Ntest_n} \quad (2.14)$$

$$nist_penalty(ratio) = \exp \left(\frac{\log(0.5)}{\log(1.5)^2} \cdot \log(ratio)^2 \right) \quad (2.15)$$

Finally, $Nmatch_weight_i$ is also a cumulative count (updated sentence by sentence), defined as:

$$Nmatch_weight_i = \sum_{n=1}^N \sum_{ngr \in S} \left(\min \left\{ N(test_n, ngr), \max_r \{ N(ref_{n,r}, ngr) \} \right\} \cdot weight(ngr) \right) \quad (2.16)$$

where $weight(ngr)$ is used to weight every n-gram according to the identity of the words it contains, expressed as follows:

$$weight(ngr) = \begin{cases} -\log_2 \left(\frac{N(ngr)}{N(mgr)} \right) & \text{if mgr exists;} \\ -\log_2 \left(\frac{N(ngr)}{Nwords} \right) & \text{otherwise;} \end{cases} \quad (2.17)$$

where mgr is the same N-gram of words contained in ngr except for the last word. $N(ngram)$ is the number of occurrences of the N-gram $ngram$ in the reference sets. $Nwords$ is the total number of words of the reference sets.

The NIST score is a quality score ranging from 0 to (worst translation) to an unlimited positive value. In practice, this score ranges between 5 or 12, depending on the difficulty of the task (languages involved and test set length).

From its definition, we can conclude that NIST favours those translations that have the same length as the average reference translation. If the provided translation is perfect but 'short' (for example, it is the result of choosing the shortest reference for each sentence), the resultant NIST score is much lower than another translation with a length more similar to that of the average reference.

2.4.1.3 mWER

Word Error Rate (WER) is a standard speech recognition evaluation metric, where the problem of multiple references does not exist. For translation, its multiple-reference version (mWER) is computed on a sentence-by-sentence basis, so that the final measure for a given corpus is based on the cumulative WER for each sentence. This is expressed in 2.18:

$$mWER = \frac{\sum_{n=1}^N WER_n}{\sum_{n=1}^N Avg_Ref_Length_n} \cdot 100 \quad (2.18)$$

where N is the number of sentences to be evaluated. Assuming we have R different references for each sentence, the *average reference length* for a given sentence n is defined as:

$$Avg_Ref_Length_n = \frac{\sum_{r=1}^R Length(Ref_{n,r})}{R} \quad (2.19)$$

Finally, the *WER* cost for a given sentence n is defined as:

$$WER_n = \min_r LevDist(Test_n, Ref_{n,r}) \quad (2.20)$$

where *LevDist* is the Levenshtein Distance between the test sentence and the reference being evaluated, assigning an equal cost of 1 for deletions, insertions and substitutions. All lengths are computed in number of words.

mWER is a percentual error metric, thus defined in the range of 0 to 100, 0 meaning the *perfect* translation (matching at least one reference for each test sentence).

From mWER description, we can conclude that the score tends to slightly favour shorter translations to longer translations. This can be explained by considering that the absolute number of errors (found as the Levenshtein distance) is being divided by the average sentence length of the references, so that a mistake of one word with respect to a long reference is being over-weighted in contrast to one mistake of one word with respect to a short reference.

Suppose we have three references of length 9, 11 and 13 (avglen = 11). If we have a translation which is equal to the shortest reference, except by one mistake, we have a score of 1/11 (where, in fact, the error could be considered higher, as it is one mistake over 9 words, that is 1/9).

2.4.1.4 mPER

Similar to WER, the so-called Position-Independent Error Rate (mPER) is computed on a sentence-by-sentence basis, so that the final measure for a given corpus is based on the cumulative PER for each sentence. This is expressed thus:

$$mPER = \frac{\sum_{n=1}^N PER_n}{\sum_{n=1}^N Avg_Ref_Length_n} \cdot 100 \quad (2.21)$$

where N is the number of sentences to be evaluated. Assuming we have R different references for each sentence, the *average reference length* for a given sentence n is defined as in equation 2.19.

Finally, the *PER* cost for a given sentence n is defined as:

$$PER_n = \min_r (Pmax(Test_n, Ref_{n,r})) \quad (2.22)$$

where $Pmax$ is the maximum between:

- POS = num. of words in the REF that are not found in the TST sent. (*recall*)
- NEG = num. of words in the TST that are not found in the REF sent. (*precision*)

in this case, the number of words includes repetitions. This means that if a certain word appears twice in the reference but only once in the test, then POS=1.

2.4.1.5 METEOR

The Metric for Evaluation of Translation with Explicit ORdering (**METEOR**) was designed to explicitly address the weaknesses in BLEU (see [Ban05]). It also produces good correlation with human judgment at the sentence or segment level, what differs from the BLEU metric in that BLEU seeks correlation at the corpus level.

It evaluates a translation by computing a score based on explicit word-to-word matches between the translation and a reference translation. If more than one reference translation is available, the given translation is scored against each reference independently, and the best score is reported.

The metric is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision. It also has several features that are not found in other metrics, such as stemming and synonymy matching, along with the standard exact word matching.

The algorithm first creates an alignment between the two given sentences, the translation output and the reference translation. The alignment is a set of mappings between unigrams. Every unigram in the translation output must map to zero or one unigram in the reference translation and vice versa. In any alignment, a unigram in one sentence cannot map to more than one unigram in another sentence.

Alignments are created incrementally in different stages, which are controlled by modules. A module is simply a matching algorithm. Matching algorithms may employ synonyms (using WordNet), stems or exact words.

Each stage is composed of two phases:

- In the first phase, all possible mappings are collected for the module being used in the stage.
- In the second phase, the largest subset of these mappings is selected to produce an alignment as defined above. If there are two alignments with the same number of mappings, the alignment is chosen with the fewest crosses, that is, with fewer intersections of two mappings.

Stages are run consecutively. Each stage adds to the final alignment those unigrams which have not been matched in previous stages. Once the final alignment is computed, the score is computed as follows:

Unigram precision P is calculated as $P = m/w_t$

where m is the number of unigrams in the translation output that are also found in the reference translation, and w_t is the number of unigrams in the translation output.

Unigram recall R is computed as $R = m/w_r$

where m is as for P , and w_r is the number of unigrams in the reference translation.

Precision and recall are combined using the harmonic mean in the following way, with recall weighted 9 times more than precision:

$$F_{mean} = \frac{10PR}{R + 9P} \quad (2.23)$$

So far, the measure only accounts for matchings with respect to single words. In order to take larger segments into account, longer N -gram matches are used to compute a penalty p for the alignment. The more mappings there are that are not adjacent in the reference and the translation output sentence, the higher the penalty will be.

In order to compute this penalty, unigrams are grouped into the fewest possible chunks (adjacent unigrams in the hypothesis and in the reference). The longer the adjacent mappings between the hypothesis and the reference, the fewer chunks there are. A translation that is identical to the reference will give just one chunk.

The penalty p is computed as $p = 0.5(c/u_m)^3$

where c is the number of chunks, and u_m is the number of unigrams that have been mapped. The final score for a segment is calculated as:

$$METEOR = F_{mean}(1 - p) \quad (2.24)$$

The penalty has the effect of reducing the F_{mean} by up to 50% if there are no bigram or longer matches.

To calculate a score over a whole corpus, or collection of segments, the aggregate values for P , R and p are taken and then combined using the same formula.

2.4.1.6 Other evaluation metrics

Apart from these, several other automatic evaluation measures comparing hypothesis translations against supplied references have been introduced, claiming good correlation with human intuition. Although not used in this Ph.D. dissertation, here we refer to some of them.

- Geometric Translation Mean, or **GTM**, measures the similarity between texts by using a unigram-based F-measure, as presented in [Tur03]

- Weighted N-gram Model, or **WNM**, introduced in [Bab04], is a variation of BLEU which assigns different value for different n-gram matches
- **ORANGE** ([Lin04b]) uses unigram co-occurrences and adapts techniques from automatic evaluation of text summarization, as presented in the ROUGE score ([Lin04a])
- **mCER** is a simple multiple-reference character error rate, and is supplied by ELDA
- As a result from a 2003 John Hopkins University workshop on Confidence Estimation for Statistical MT, [Bla04] introduce evaluation metrics such as Classification Error Rate (**CER**) or the Receiving Operating Characteristic (**ROC**)
- From a more intuitive point of view, in [Sno05] Translation Error Rate, or **TER**, is presented. This measures the amount of editing that a human would have to perform to change a system output so it exactly matches a reference translation. Its application in real-life situation is reported in [Prz06].

Finally, in [Gim06] the IQMT framework is presented. This tool follows a 'divide and conquer' strategy, so that one can define a set of metrics and then combine them into a single measure of MT quality in a robust and elegant manner, avoiding scaling problems and metric weightings.

2.4.2 Human Metrics

Human evaluation metrics require a certain degree of human intervention in order to obtain the quality score. This is a very costly evaluation strategy that seldom can be conducted. However, thanks to international evaluation campaigns, these measures are also used in order to compare different systems.

Usually, the tendency has been to evaluate *adequacy* and *fluency* (or other relevant aspects of translation) according to a 1 to 5 quality scale. Fluency indicates how natural the hypothesis sounds to a native speaker of the target language, usually with these possible scores: 5 for Flawless, 4 for Good, 3 for Non-native, 2 for Disfluent and 1 for Incomprehensible. On the other hand, Adequacy is assessed after the fluency judgment is done, and the evaluator is presented with a certain reference translation and has to judge how much of the information from the original translation is expressed in the translation by selecting one of the following grades: 5 for all of the information, 4 for most of the information, 3 for much of the information, 2 for little information, and 1 for none of it³.

However, another trend is to manually post-edit the references with information from the test hypothesis translations, so that differences between translation and reference account only for errors and the final score is not influenced by the effects of synonymy. The human targeted reference is obtained by editing the output with two main constraints, namely that the resultant references preserves the meaning and is fluent.

In this case, we refer to the measures as their human-targeted variants, such as HBLEU, HMETEOR or HTER as in [Sno05]. Unfortunately, this evaluation technique is also costly and cannot be used constantly to evaluate minor system improvements. Yet we are of the opinion

³These grades are just orientative, and may vary depending on the task.

that, in the near future, these methods will gain popularity do to the fact that, apart from providing a well-founded absolute quality score, they produce new reference translations that can serve to automatically detect and classify translation errors.

Regarding automatic error classification or analysis, some recent works on the subject suggest that it is possible to use linguistic information to automatically extract further knowledge from translation output than just a single quality score (we note the work of [Pop06a,Pop06b]).

Chapter 3

N-gram-based approach to Statistical Machine Translation

This chapter is devoted to the study of the *N*-gram-based approach to SMT, with special emphasis on the bilingual *N*-gram translation model, core model of the UPC SMT system. The system incorporates a set of additional models in the form of a log-linear combination of feature functions. Hence, the core translation model is extended with complementary information. The chapter is organized as follows:

- Firstly, the translation model is discussed in §3.2. The implementation details and estimation in the form of a standard *N*-gram language model.
- The mathematical framework underlying the log-linear combination of models is presented in §3.3. Each additional feature model is also described, along with relevant decoding, training and optimization details.
- §3.4 reports on the experiments conducted in order to evaluate the impact in translation quality of the different system elements using a large sized Spanish-English data translation task. It also includes a manual error analysis to identify the most important shortcomings of the system.
- §3.5 provides a detailed comparison of the studied system to a standard phrase-based system. The comparison includes modeling and translation unit singularities as well as a performance comparison under different data size conditions.
- In §3.6 a summary of the chapter can be found, highlighting the main conclusions extracted from it.

3.1 Introduction

The translation system described in this thesis work implements a translation model that has been derived from the finite-state perspective. More specifically, from the work in [Cas01] and [Cas04], where the translation model is implemented by using a finite-state transducer.

However, in the system presented here, the translation model is implemented by using *N*-grams. In this way, the proposed translation system can take full advantage of the smoothing and consistency provided by standard back-off *N*-gram models.

3.2 Bilingual *N*-gram Translation Model

As already mentioned, the translation model implemented by our SMT system is based on bilingual *N*-grams. This model constitutes actually a language model of a particular “*bi-language*” composed of bilingual units (translation units) which are referred to as **tuples**. In this way, the translation model probabilities at the sentence level are approximated by using *N*-grams of tuples, such as described by the following equation:

$$p_{BM}(s_1^J, t_1^I) \approx \prod_{k=1}^K p((s, t)_k | (s, t)_{k-1}, (s, t)_{k-2}, \dots, (s, t)_{k-n+1}) \quad (3.1)$$

where *t* refers to target, *s* to source and $(s, t)_k$ to the k^{th} tuple of a given bilingual sentence pair. It is important to notice that, given that both languages are linked up in tuples, the context information provided by this translation model is bilingual.

As any standard *N*-gram language model, our translation model is estimated over a training corpus composed of sentences of the language being modeled. In this case, sentences of the “*bi-language*” previously introduced. Next, we detail the method employed to transform a word-to-word aligned training corpus into the tuples training corpus needed to feed the *N*-gram language model.

3.2.1 From Word-alignments to Translation Units

Translation units (tuples in our case) consist of the core elements of any SMT system. So far the translation process is mainly carried out by composing these small pieces, the likelihood of obtaining accurate translations highly depends on the availability of ‘*good*’ units. In consequence, the extraction of these units is a key process when building a SMT system.

From a conceptual point of view, the final goal of the tuple extraction process is to obtain a set of units with a high level of translation accuracy (*i.e.* the source/target sides of a translation unit consist of translations of each other) and with the ‘re-usability’ capacity (re. that they can be recycled in order to produce valid translations in certain unseen situations, the more the better).

Tuples are generated as a segmentation of each pair of training sentences. This segmentation allows the estimation of the *N*-gram probabilities appearing in Equation 3.1). Figure 3.1 illus-

trates the importance of the tuple extraction process. It shows three different segmentations for the sentence pair '*Maria finalmente abofeteó a la bruja # Maria finally slapped the witch*'.

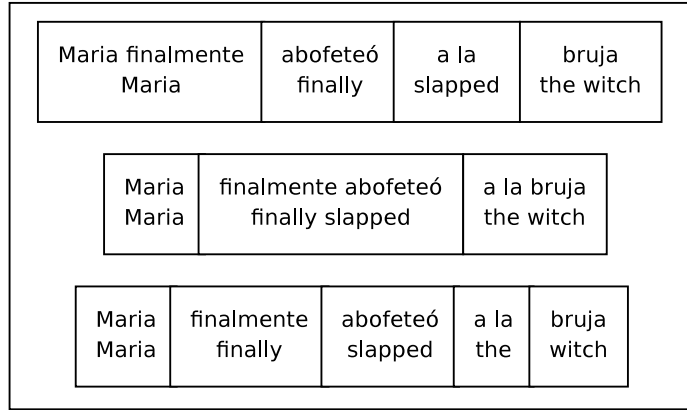


Figure 3.1: Three tuple segmentations of the sentence pair: '*Maria finalmente abofeteó a la bruja # Maria finally slapped the witch*'.

Four tuples are generated following the first segmentation (top). They have a very low level of translation accuracy. Translation of new sentences using these tuples can only succeed when translating the same source sentence that originated the units (thus, we get the lowest re-usability capacity). Considering the second segmentation (middle), the resulting tuples can be considered accurate in terms of translation between their source and target sides. However, the re-usability capacity is not as high as desired. For instance, the second tuple *finally_slapped#finalmente_abofeteó* can only be used if both words: *finally* and *slapped* appear together when translating new sentences. Finally, the third segmentation (bottom) shows apparently the best values of translation accuracy and re-usability for their constituent units.

3.2.1.1 Tuple Segmentation

Tuples are typically extracted from a word-to-word aligned corpus in such a way that a unique segmentation of the bilingual corpus is achieved. Although in principle, any Viterbi alignment should facilitate tuple extraction, the resulting tuple vocabulary highly depends on the particular alignment set considered, and so the translation results.

Different from other implementations, where one-to-one [Ban00a] or one-to-many [Cas04] alignments are used, tuples are typically extracted from many-to-many alignments.

This implementation produces a monotonic segmentation of bilingual sentence pairs, which allows simultaneously capturing contextual and reordering information into the bilingual translation unit structures. This segmentation is used to estimate the N -gram probabilities appearing in (3.1).

In order to guarantee a unique segmentation of the corpus, tuple extraction is performed according to the following constraints [Cre04]:

- a monotonic segmentation of each bilingual sentence pair is produced,

- no word inside a tuple can be aligned to words outside the tuple, and
- no smaller tuples can be extracted without violating the previous constraints.

Notice that according to this, tuples can be formally defined as the set of shortest phrases (introduced in [Zen02]) that provides a monotonic segmentation of the bilingual corpus.

Figure 3.2 presents a simple example illustrating the unique tuple segmentation for a given pair of word-to-word aligned sentences.

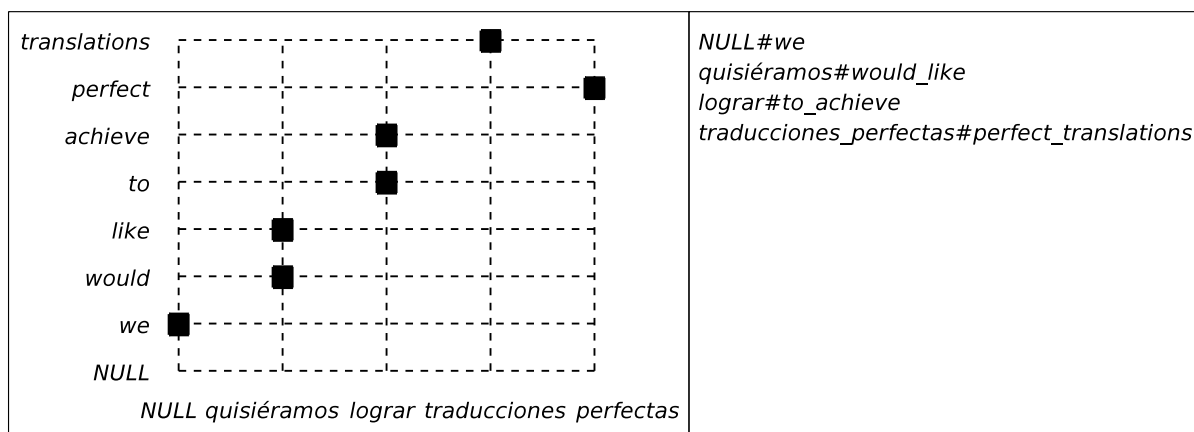


Figure 3.2: *Tuple extraction from a word-to-word aligned pair of sentences.*

According to our experience, the best performance is achieved when the union [Och03c] of the source-to-target and target-to-source alignment sets (IBM models [Bro93]) is used as starting point of the tuple extraction. Additionally, the use of the union can also be justified from a theoretical point of view by considering that the union set typically exhibits higher recall values than other alignment sets such as the intersection and source-to-target.

Figure 3.3 illustrates the extraction of translation units following two different word alignments, the union and the intersection of the source-to-target and target-to-source alignment sets. Intersection and union alignments are drawn using respectively black and unfilled boxes. As it can be seen, the set of translation units extracted from each alignment set are remarkably different from each other. The suitability of each alignment set is tightly coupled with the pair of languages considered in translation.

3.2.1.2 Source-NULled Tuples

Following the tuple definition, a unique sequence of tuples is extracted for each training pair of sentences and using the corresponding word-to-word alignment. However, the resulting sequence of tuples may fall into a situation where the sequence needs to be further refined.

It frequently occurs that some target words linked to NULL end up producing tuples with NULL source sides. Consider for example the first tuple of the example presented in figure 3.2. In this example, “*NULL#we*” is a source-NULLed tuple if Spanish is considered to be the source language.

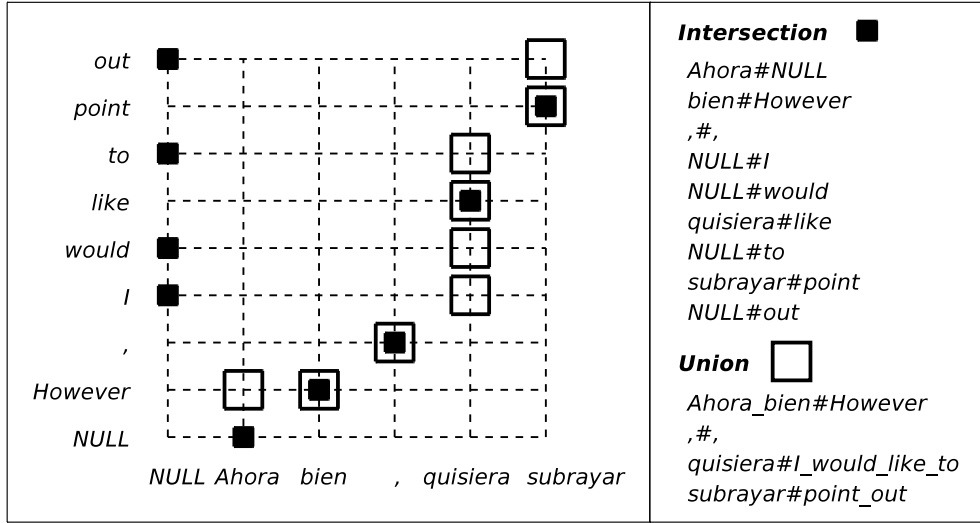


Figure 3.3: Tuple extraction from a certain word aligned pair of sentences.

In order to re-use these units when decoding new sentences, the search should allow the apparition (generation) of NULL input words. This is the classical solution in the finite-state transducer framework, where NULL words are referred to as “*epsilon arcs*” [Kni98, Ban00b]. However, “*epsilon arcs*” significantly increase the decoding complexity and are not implemented in our decoder. Therefore, source-NULled units are not allowed, and a certain hard decision must be taken to avoid the apparition of these units.

In our system implementation, this problem is easily solved by preprocessing the set of alignments before extracting tuples, in such a way that any target word that is linked to NULL is attached to either its precedent word or its following word. In this way, no target word remains linked to NULL, and source-NULled tuples will not occur during tuple extraction.

In the example of figure 3.2 this decision is straightforward taken as no previous tuple exist. Thus the resulting refined segmentation contains the tuple *quisieramos#we_would_like*. However, when both, the previous and the next tuples exist, the decision should be taken towards maximizing the accuracy and usability of the resulting tuples.

So far three segmentation strategies to solve the source-NULled units problem have been proposed:

- The first is a very simple approach consisting of to attach always the target words involved in source-NULled tuples to the following tuple (**always NEXT**). When no tuple appears next, the previous one is used instead. This approach was first introduced in [Gis04].

Apart from simplicity and extreme efficiency, we do not observe any other advantage of this approach, which on the other hand does not follow any linguistic or statistical criterion.

- The second strategy considers the goal of obtaining the set of units with highest translation accuracy. It employs a word-based lexicon model to compute a translation probability of the resulting tuples given the two competing situations. This approach (**LEX model weight**) was first introduced in [Cre05a]. The weight for each tuple is defined as:

$$\frac{1}{I} \prod_{j=1}^J \sum_{i=0}^I p_{LEX}(t_i|s_j) p_{LEX'}(s_j|t_i) \quad (3.2)$$

where s and t represent source and target sides of a tuple, I and J their respective number of words and LEX (and LEX') consist of the lexicon model (estimated in both directions, source-to-target and target-to-source). Typically, *IBM Model 1* probabilities are used as lexicon models.

Many source-NULled words represent articles, prepositions, conjunctions and other particles whose main function is to ensure the grammatical correctness of a sentence, complementing other more informative words. Therefore, their probabilities to translate to another word are not very meaningful.

- The third approach considers that the ideal tuple segmentation strategy should take a global decision for each source-NULled unit attempting to obtain the set of tuples and N -grams which better represent the unseen universe of events, meaning the one with less entropy.

From a linguistic point of view, one can regard the tuple segmentation problem around source-NULled words as a monolingual decision related to whether a given target word is more connected to its preceding or following word.

Intuitively, we can expect that a good criterion to perform tuple segmentation lays in preserving grammatically-connected phrases (such as, for instance, articles together with the noun they precede) in the same tuple, as this may probably lead to a simplification of the translation task. On the contrary, splitting linguistic units into separate tuples will probably lead to a tuple vocabulary increase and a higher sparseness, producing a worse (and more entropic) N -gram translation model.

This approach (**POS entropy**) is further detailed in [Gis06] where comparison experiments are also carried out considering the three different strategies for a Spanish–English translation task.

Conclusions drawn in [Gis06] mainly consist of that in principle, the **POS entropy** approach seems to be highly correlated to a human segmentation obtaining also highly translation accuracy results. However, when the N -gram translation model is log-linearly combined with additional features, and specially for large-vocabulary tasks, the impact of the segmentation employed is minimized.

3.2.1.3 Embedded-word Tuples

Another important issue regarding the N -gram translation model is related to the problem of embedded words. It refers to the fact that the tuple representation is not able to provide translations for individual words in all cases. Embedded words can become a serious drawback when they occur in relatively important amounts into the tuple vocabulary.

Consider for example the word “*translations*” in Figure 3.2. As seen from the figure, this word appears embedded into the tuple “*traducciones perfectas#perfect.translations*”. If it happens to be that a similar situation is encountered for all other occurrences of such a word in the training corpus, then, no translation probability for an independent occurrence of such a word will exist.

According to this, the problem resulting from embedded words can be partially solved by incorporating a bilingual dictionary able to provide word-to-word translation when required by the translation system.

The solution typically adopted in our system implements the following strategy for handling embedded words: First, one-word tuples for each detected embedded word are extracted from the training data and their corresponding word-to-word alignments; Then, the tuple N -gram model is enhanced by including all embedded-word tuples as unigrams into the model. The probability associated to these new unigrams is set to the same value as the one computed for the unknown word.

Since a high precision alignment set is desirable for extracting such one-word tuples and estimating their probabilities, the intersection of both alignments, source-to-target and target-to-source, is typically used instead of the union.

The use of embedded-word tuples is particularly suited for translation tasks with a relatively small amount of training material and important reordering needs. The particularities of this kind of tasks force the apparition of a larger number of embedded words. In the particular case of the EPPS tasks (described in section A.1), embedded words do not constitute a real problem because of the great amount of training material and the reduced size of the test data set. On the contrary, in other translation tasks with less available training material the embedded-word handling strategy described above has resulted to be very useful [Gis04].

This dictionary solution forces the model to fall back to an incontextual word-based translation for embedded words, which is specially negative for language pairs with strong reordering needs, where long tuples appear more often increasing the number of embedded words.

Similarly to single embedded-words, arbitrary large sequences of words can also be embedded into larger tuples. In such a case, the same solution could be adopted to account for the hidden translation options. In this work, the embedded-word strategy has only been implemented for single words.

3.2.1.4 Tuple Vocabulary Pruning

The third and last issue regarding the N -gram translation model is related to the computational cost resulting from the tuple vocabulary size during decoding. The idea behind this refinement is to reduce both computation time and storage requirements without degrading translation performance. In our N -gram based SMT system implementation, the tuple vocabulary is pruned out by using histogram counts. This pruning is performed by keeping the N most frequent tuples with common source sides.

Notice that such a pruning, since performed before computing tuple N -gram probabilities, has a direct incidence on the translation model probabilities, and then on the overall system performance. For this reason, the pruning parameter N is critical for an efficient usage of the translation system. While a low value of N will significantly decrease translation quality, on the other hand, a large value of N will provide the same translation quality than a more adequate N , but with a significant increment in computational costs. The optimal value for this parameter depends on data and is typically adjusted empirically for each considered translation task.

Given the noisy data from which word-to-word alignments and translation units are extracted, tuple pruning can also be seen as a cleaning process, where 'bad' units are discarded.

To illustrate this common situation, The next list shows the 20 more common translations of the Spanish word '*en*' collected as tuples from the Spanish-English corpus detailed in Section A.1.1.

in (274253)	NULL (120558)	on (47243)	at (22405)	into (13422)
to (12828)	in_the (7343)	within (6962)	with (5742)	of (5512)
as (4624)	where (4527)	by (3942)	for (3874)	,_in (2943)
when (2779)	under (2216)	during (1881)	over (1854)	in_a (1808)

Translations are sorted according to their number of apparitions in the entire corpus (shown in parentheses).

3.2.2 N-gram Language Model Estimation

The estimation of the special 'translation model' is carried out using the freely-available **SRILM** toolkit. First presented in [Sto02]. This collection of C++ libraries, executable programs, and helper scripts was designed to allow both production of and experimentation with statistical language models for speech recognition and other applications, supporting creation and evaluation of a variety of language model types based on N -gram statistics (including many smoothing strategies), among other related tasks.

Empirical reasons typically support the decision of using the options for Kneser-Ney smoothing [Kne95] and interpolation of higher and lower N -grams.

Figure 3.4 shows the format of a bilingual N -gram language model estimated by means of the SRILM toolkit over a training corpus expressed in the form of tuples.

As it can be seen, the model estimates the probability of unknown tokens ' $\langle unk \rangle$ '. Unknown tokens may appear in the language model uncontextualized (in the form of unigrams) as well as within a longer N -gram. The last N -gram probability of figure 3.4 shows the apparition of an unknown unit ' $\langle unk \rangle$ ' in a tuple 3-gram.

Units pruned out before the model estimation are used by the language modeling toolkit to estimate the probabilities of the ' $\langle unk \rangle$ ' token. In the example of figure 3.4, the pruned unit '*quisiera#I_would_like_to*' appears in the model as an unknown unit ' $\langle unk \rangle$ '. The N -gram ' $\langle \#, \langle unk \rangle, subrayar \#point_out \rangle$ ' is one of the 3-grams where the pruned unit is involved.

When decoding new sentences, unit N -grams containing unknown tokens (such as ' $\langle \#, \langle unk \rangle, subrayar \#point_out \rangle$ ') can be used with input sentences containing out of vocabulary words (input words which do not appear as source side of any translation unit). For instance, when translating the sentence '*... , necesito subrayar ...*', the 3-gram of the example will be used if the word '*necesito*' is an out of vocabulary word.

Tokens corresponding to the beginning and end of sentence are also taken into account in the language model ($\langle s \rangle$ and $\langle /s \rangle$). Although they lack of a translation meaning, these special tokens are also used in our translation model as being part of the bilingual translation history. Following this, equation 3.1 needs to be refined to introduce these new tokens:

$$p(s_1^J, t_1^I) \approx \prod_{k=0}^{K+1} p((s, t)_k | (s, t)_{k-1}, (s, t)_{k-2}, \dots, (s, t)_{k-n+1}) \quad (3.3)$$

where $(s, t)_0$ and $(s, t)_{K+1}$ refer respectively to tokens $\langle s \rangle$ and $\langle /s \rangle$.

However, we typically employ equation 3.1 when referring to our special N -gram translation model. Both equations can be considered equivalent when the input sentence (s_1^J) is extended to contain the beginning and ending tokens ($s_1 = \langle s \rangle$ and $s_J = \langle /s \rangle$).

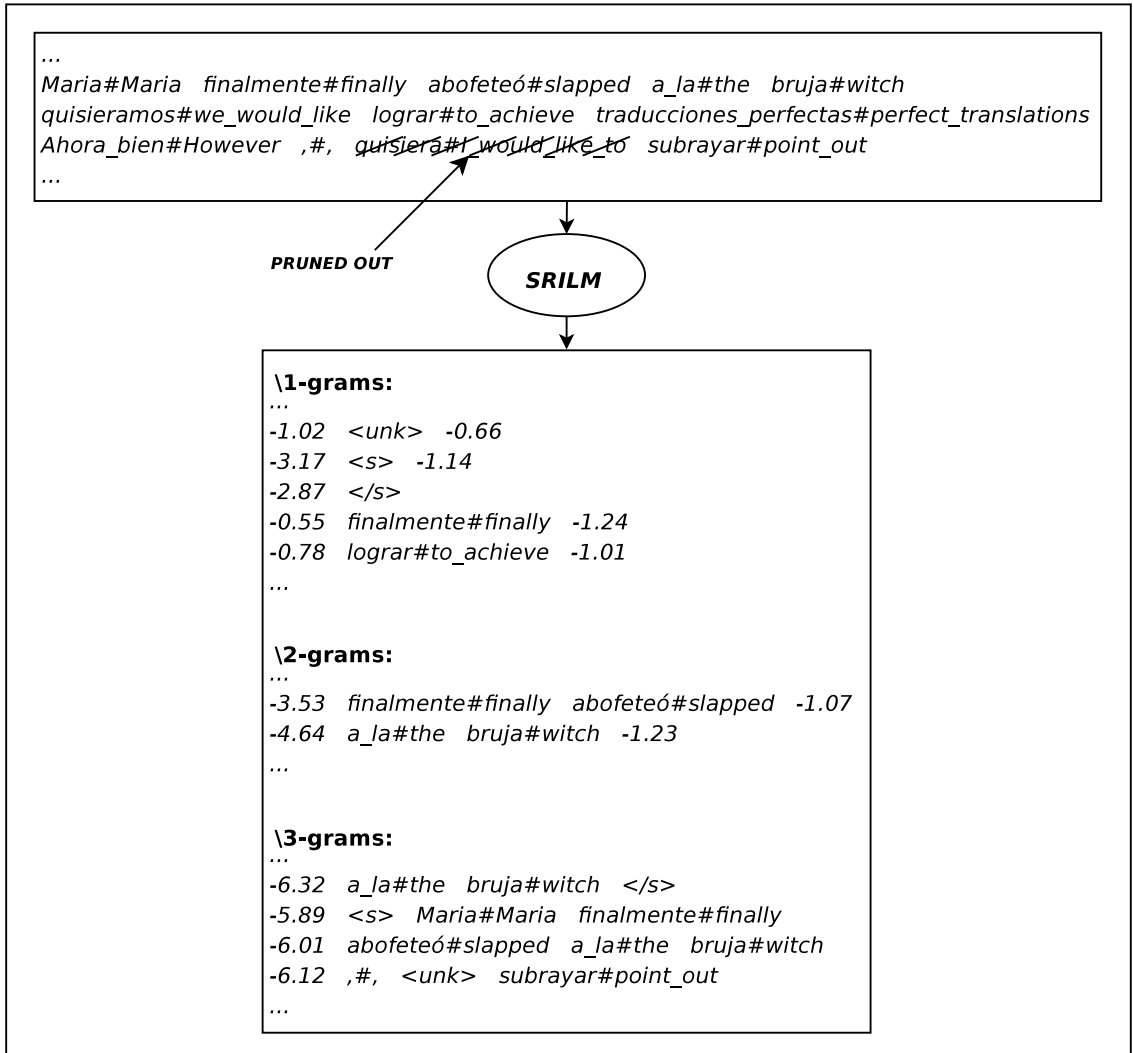


Figure 3.4: Estimation of a ‘bilingual’ N -gram language model using the SRILM toolkit.

3.3 *N*-gram-based SMT System

3.3.1 Log-linear Combination of Feature Functions

Current translation systems have replaced the original noisy channel approach by a more general approach, which is founded on the principles of maximum entropy applied to Natural Language Processing tasks [Ber96]. Under this framework, given a source sentence s , the translation task is defined as finding that target sentence t which maximizes a log-linear combination of multiple feature functions $h_i(s, t)$, as described by the following equation (equivalent to equation 2.6):

$$\hat{t} = \underset{t}{\operatorname{argmax}} \sum_m \lambda_m h_m(s, t) \quad (3.4)$$

where λ_m represents the coefficient of the m^{th} feature function $h_m(s, t)$, which corresponds to a log-scaled version of m^{th} -model probabilities. Optimal values for the coefficients λ_m s are estimated via an optimization procedure on a certain development data set.

In addition to the bilingual *N*-gram translation model, the *N*-gram based SMT system implements four feature functions which provide complementary views of the translation process, namely a target language model, a word bonus model and two lexicon models. These features are described next.

3.3.1.1 Target *N*-gram Language Model

This feature provides information about the target language structure and fluency, by favoring those partial-translation hypotheses which are more likely to constitute correctly structured target sentences over those which are not. The model implements a standard word *N*-gram model of the target language, which is computed according to the following expression:

$$p_{LM}(s_1^J, t_1^I) \approx \prod_{i=1}^I p(t_i | t_{i-N+1}, \dots, t_{i-1}) \quad (3.5)$$

where t_i refers to the i^{th} target word.

From a theoretical point of view, the translation bilingual model already constitutes a source and target language model. Therefore, one could be led to think that this target language model is redundant and unnecessary. However, the bilingual model is more liable to suffer from sparseness than any monolingual model, which can turn this model helpful whenever tuple *n*-grams are not well estimated.

3.3.1.2 Word Bonus

The use of any language model probabilities is associated with a length comparison problem. In other words, when two hypotheses compete in the search for the most probable path, the one using less number of elements (being words or translation units) will be favored against the one using more. The accumulated partial score is computed by multiplying a different number of

probabilities. This problem results from the fact that the number of target words (or translation units) used for translating a test set is not fixed and equivalent in all paths.

The word bonus model is used in order to compensate the system preference for short target sentences. It is implemented following the next equation:

$$p_{WB}(s_1^J, t_1^I) = \exp(I) \quad (3.6)$$

where I consists of the number of target words of a translation hypothesis.

3.3.1.3 Source-to-target Lexicon Model

This feature actually constitutes a complementary translation model. This model provides, for each tuple, a translation probability estimate between the source and target sides of it. This feature is implemented by using the IBM-1 lexical parameters [Bro93, Och04a]. According to this, the source-to-target lexicon probability is computed for each tuple according to the following equation:

$$p_{LEX_{st}}(s_1^J, t_1^I) = \log \frac{1}{(I+1)^J} \prod_{j=1}^J \sum_{i=0}^I q(t_j^n | s_i^n) \quad (3.7)$$

where s_i^n and t_j^n are the i^{th} and j^{th} words in the source and target sides of tuple $(t, s)_n$, being I and J the corresponding total number of words in each side of it. In the equation $q(\cdot)$ refers to IBM-1 lexical parameters which are estimated from alignments computed in the source-to-target direction.

3.3.1.4 Target-to-source Lexicon Model

Similar to the previous feature, this feature function constitutes a complementary translation model too. It is computed exactly in the same way the previous model is, with the only difference that IBM-1 lexical parameters are estimated from alignments computed in the target-to-source direction instead.

$$p_{LEX_{ts}}(s_1^J, t_1^I) = \log \frac{1}{(J+1)^I} \prod_{i=1}^I \sum_{j=0}^J q(s_i^n | t_j^n) \quad (3.8)$$

where $q(\cdot)$ refers in this case to IBM-1 lexical parameters estimated in the target-to-source direction.

3.3.2 Training Scheme

Training an *N*gram-based SMT system as described in the previous lines can be graphically represented as in Figure 3.5.

The first preliminary step requires the preprocessing of the parallel data, so that it is sentence aligned and tokenized. By sentence alignment the division of the parallel text into sentences and the alignment from source sentences to target sentences is referred.

By tokenization, we refer to separating punctuation marks, classifying numerical expressions into a single token, and in general, simple normalization strategies tending to reduce vocabulary size without an information loss (*i.e.* which can be reversed if required). Additionally, further tokenization can be introduced to complex morphology languages (such as Spanish or Arabic) in order to reduce the data sparseness problem and/or mimic source and target number of words (*i.e.* contractions as *del* or *al* are split into *de el* and *a el* when translating into/from the English of *the* and *to the*).

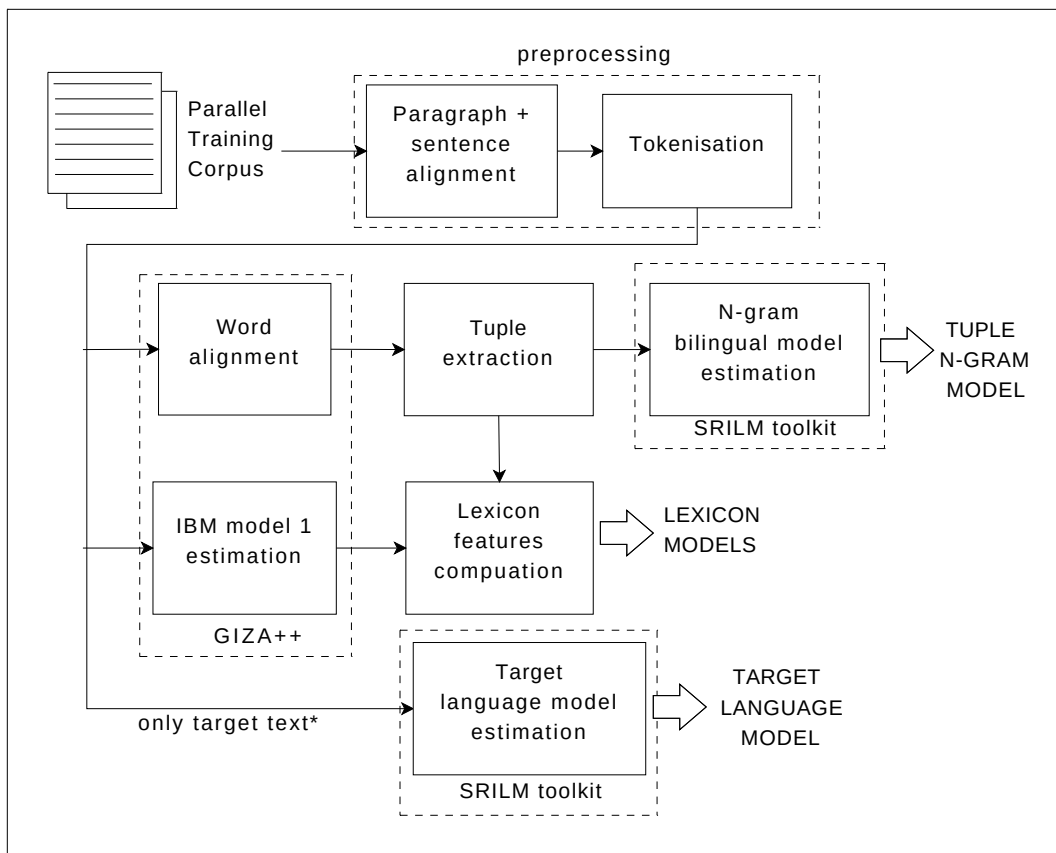


Figure 3.5: Feature estimation of an N-gram-based SMT system from parallel data. Flow diagram.

Then, word alignment is performed, by estimating IBM translation models (see §2.1.1) from parallel data and finding the Viterbi alignment in accordance to them. This process is typically carried out using the GIZA toolkit (see §2.1.1). However, any alignment toolkit can be used if it ends up producing word-to-word alignments.

Before estimating the bilingual N-grams, a tuple extraction from word-aligned data needs to be done. The tuple extraction process includes also the refinement methods detailed in section 3.2.1.

Additional training blocks include estimating a monolingual language model with the target language material only (which could be extended with monolingual data, if available) and computing the two aforementioned lexicon models from lexicon model probabilities (typically IBM model 1).

3.3.3 Optimization Work

We Typically train our models according to an error-minimization function on a certain development data, as discussed in §2.2.1. In our *N*gram-based SMT system, this process assigns the λ_m weights of each feature function shown in equation 3.4. Optimal log-linear coefficients are estimated via the optimization procedure described next. First, a development data set which overlaps neither the training set nor the test set is required. Then, translation quality over the development set is maximized by iteratively varying the set of coefficients. This optimization procedure is performed by using an in-house developed tool, which is based on a SIMPLEX method [Nel65]. The optimization process is graphically illustrated in Figure 3.6.

It can be divided into an external and an internal loop. In the external loop, a limited number of translations over the development set are carried out. Each translation is performed with new values for the set of λ_m weights (values are refined in the internal loop) producing an *N*-best list. The external loop ends when a maximum number of translations is achieved or when no accuracy improvement is seen in the last translation.

The internal loop aims at finding the best translation of the *N*-best list by tuning the values of the λ_m weights (coefficient's refinement). It consists of a re-scoring process that employs the same models used in the overall search. The internal optimization is based on the SIMPLEX [Nel65] algorithm.

The adequacy of this optimization process is founded on the following assumptions:

- There exists a set (or sets) of weights maximizing the score in the development set, and it can be found
- The weights maximizing the score on the development set will maximize the score on the test set (unless over-fitting problems)
- Maximizing the score produces better translations (which is related to the correlation between automatic and manual evaluation metrics)

Additionally, the double loop procedure assumes that a translation of the development set (external loop) is computationally more expensive (in terms of decoding time) than the re-scoring process performed in the *N*-best list (internal loop).

Translation is carried out using the MARIE decoder. Being a major contribution of this thesis work, Chapter 5 is entirely dedicated to define and discuss the decoder details.

3.4 Experiments

In this section we conduct a set of experiments aiming at evaluating the adequacy of the different system elements detailed in the previous sections. It is worth mentioning that conclusions drawn

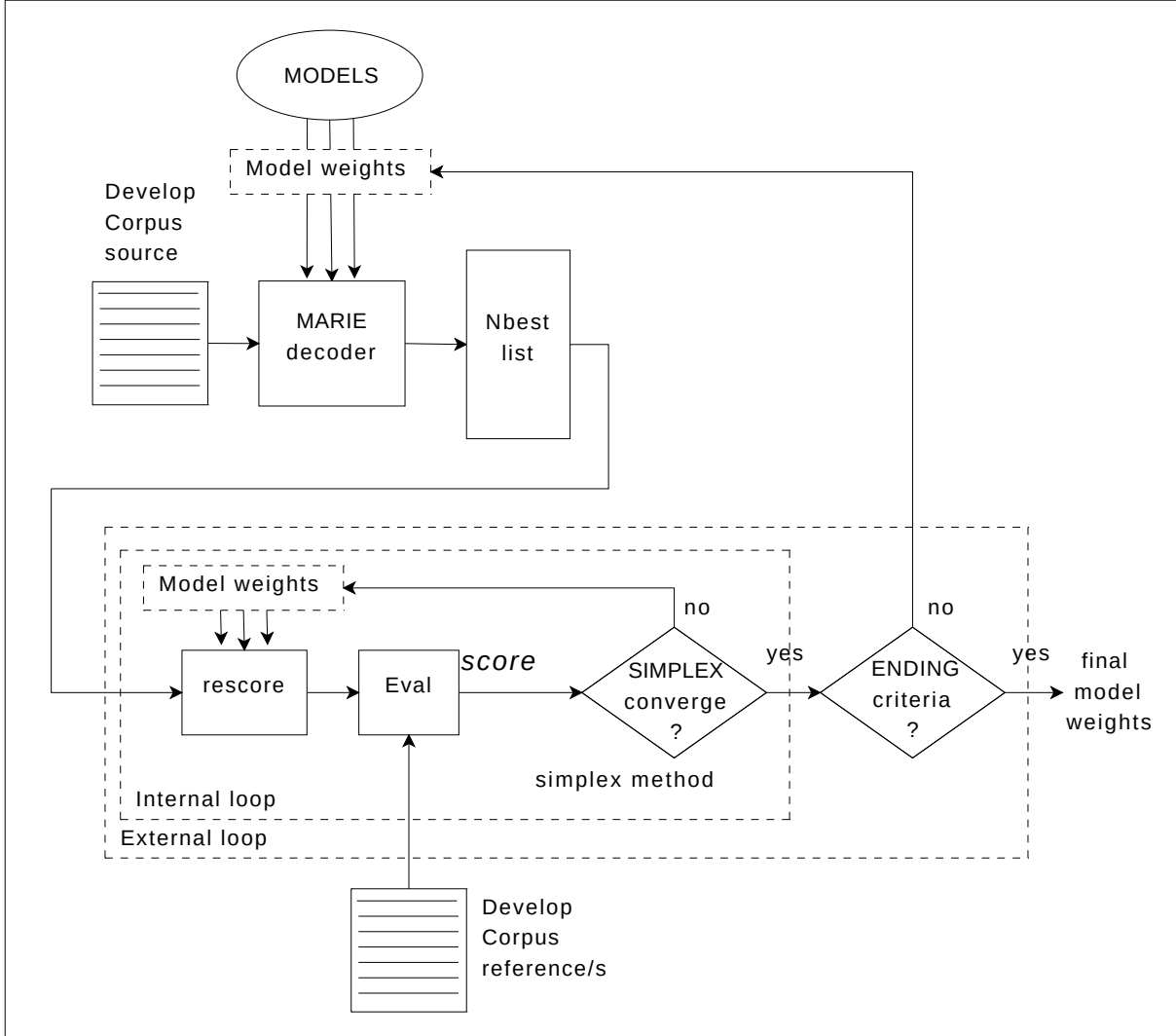


Figure 3.6: Optimization procedure. Flow diagram.

from the experiments are highly constrained to the context where they have been obtained, *i.e.* the translation task and data conditions employed.

In this case, all the experiments have been carried out over a large data sized Spanish-English corpus, detailed in Section A.1.1.

Word-to-word alignments are performed in both directions, source-to-target and target-to-source, using the GIZA++ [Och00b] toolkit. A total of five iterations for models IBM1 and HMM, and three iterations for models IBM3 and IBM4, are performed. Then, the obtained alignment sets are used for computing the refine, intersection and union sets of alignments from which translation units are extracted. The same decoder settings are used for all system optimizations. They consist of the following:

- decoding is performed monotonically, *i.e.* no reordering capabilities are used,

- although available in the decoder, threshold pruning is never used, and
- a histogram pruning of 50 hypotheses is always used.

Four experimental settings are considered in order to evaluate the relative contribution of different system elements to the overall performance of the N -gram-based translation system. For each setting, the impact on translation quality of a system parameter is evaluated, namely: tuple extraction and pruning, N -gram models size, source-NULled tuple strategy and feature function contribution. The standard system configuration is defined in terms of the following parameters:

- Alignment set used for tuple extraction: UNION
- Tuple vocabulary pruning parameter
 - Spanish-to-English: $N = 20$
 - English-to-Spanish: $N = 30$
- N -gram size used in translation model: 3
- N -gram size used in target-language model: 3
- Expanded translation model with embedded-word tuples: YES
- source-NULled tuple handling strategy: always NEXT
- Feature functions considered: target LM, word bonus, source-to-target lexicon and target-to-source lexicon

In the four experimental settings considered, which are presented in the following sections, a total amount of 7 different system configurations are evaluated in both translation directions, English-to-Spanish and Spanish-to-English. Hence, a total amount of 14 different translation experiments are performed. For each of these cases, the corresponding test set is translated by using the corresponding estimated models and set of optimal coefficients. Translation results are evaluated in terms of mWER and BLEU by using the two references available for each language test set.

3.4.1 Tuple Extraction and Pruning

As introduced in Section 3.2, a tuple set for each translation direction is extracted from a given alignment set. Afterwards, source-NULled tuples are avoided, and the resulting vocabulary of units is pruned out to finally estimate an N -gram language model.

Tables 3.1 and 3.2 present model size and translation accuracy for the tuple N -gram model when tuples are extracted from different alignment sets and when different pruning parameters are used, respectively. Translation accuracy is measured in terms of the BLEU [Pap02] and mWER score, which are computed here for translations generated by using the tuple N -gram model **alone** (in the case of table 3.1), and by using the standard system described in the beginning of section 3.4 (in the case of table 3.2). Both translation directions, Spanish-to-English and English-to-Spanish, are considered in both tables.

In the case of table 3.1, model size and translation accuracy are evaluated against the type of alignment set used for extracting tuples. Three different alignment sets are considered: **source-to-target**, the **union** of source-to-target and target-to-source, and the **refined** alignment method described in [Och03c]. A pruning parameter value of $N = 20$ was used for the Spanish-to-English direction, while a value of $N = 30$ was used for the English-to-Spanish direction. Tuple vocabulary sizes and their corresponding number of N -grams (in millions), and translation accuracy when tuples are extracted from different alignment sets are shown. Notice that BLEU and mWER measurements in this table correspond to translations computed by using the tuple N -gram model alone.

Table 3.1: *Model size and translation accuracy derived of the alignment set used to extract translation units.*

Direction	Alignment set	Tuple voc.	bigrams	trigrams	mWER	BLEU
ES $\rightarrow$ EN	source-to-target	1.920	6.426	2.353	40.94	.4424
	union	2.040	6.009	1.798	39.71	.4745
	refined	2.111	6.851	2.398	40.24	.4594
EN $\rightarrow$ ES	source-to-target	1.813	6.263	2.268	44.61	.4152
	union	2.023	6.092	1.747	44.46	.4276
	refined	2.081	6.920	2.323	44.39	.4193

As clearly seen from table 3.1, the union alignment set happens to be the most favorable one for extracting tuples in both translation directions since it provides a significantly better translation accuracy, in terms of the translation scores, than the other two alignment sets considered. Notice also that the union set is the one providing the smallest model sizes according to the amount of bigrams and trigrams. This might explain the improvement observed in translation accuracy, with respect to the other two cases, in terms of model sparseness.

In the case of table 3.2, model size (tuple vocabulary sizes and their corresponding number of N -grams in millions) and translation accuracy are compared for three different pruning conditions: $N = 30$, $N = 20$ and $N = 10$. The system parameters are those described in the beginning of Section 3.4.

Table 3.2: *Model size and translation accuracy derived of the tuple vocabulary pruning.*

Direction	Pruning	Tuple voc.	bigrams	trigrams	mWER	BLEU
ES $\rightarrow$ EN	$N = 30$	2.109	6.233	1.805	34.89	.5440
	$N = 20$	2.040	6.009	1.798	34.94	.5434
	$N = 10$	1.921	5.567	1.759	35.05	.5399
EN $\rightarrow$ ES	$N = 30$	2.023	6.092	1.747	40.34	.4688
	$N = 20$	1.956	5.840	1.733	41.29	.4671
	$N = 10$	1.843	5.342	1.677	41.81	.4595

Notice from table 3.2 how translation accuracy is clearly affected by pruning. In the case of Spanish-to-English, values of $N = 20$ and $N = 10$, while providing a tuple vocabulary reduction of 3.27% and 8.91% with respect to $N = 30$, respectively, produce a translation BLEU reduction of 0.11% and 0.75% (similar results are achieved in terms of mWER).

On the other hand, in the case of English-to-Spanish, values of $N = 20$ and $N = 10$ provide a

tuple vocabulary reduction of 3.31% and 8.89% and a translation BLEU reduction of 0.36% and 1.98% with respect to $N = 30$, respectively (similar results in terms of mWER). According to these results, a similar tuple vocabulary reduction seems to affect English-to-Spanish translations more than it affects Spanish-to-English translations. For this reason, we typically adopt $N = 20$ and $N = 30$ as the best pruning parameter values for Spanish-to-English and English-to-Spanish, respectively.

Apart from the considered effect on translation accuracy, the tuple vocabulary pruning produces also an important influence on the efficiency of the global search. In Section 5.2.5 is shown an upper bound estimation of the efficiency of the search, where the vocabulary of units plays a significant role.

An important observation derived from table 3.2 is the higher values of translation BLEU (and lower in terms of mWER) with respect to the ones presented in table 3.1. This is because, as mentioned above, results presented in table 3.2 were obtained by considering a full translation system which implements the tuple N -gram model along with the additional four feature functions described in Section 3.3.1. The relative impact of described feature functions on translation accuracy is studied in detail in Section 3.4.4.

3.4.2 Translation and Language N -gram Size

After tuple pruning, an N -gram model is estimated for each translation direction by using the SRI Language Modeling toolkit. The options for Kneser-Ney smoothing [Kne95] and interpolation of higher and lower N -grams are typically used. Similarly, a word N -gram target language model is estimated for each translation direction by using the same toolkit. Again, as in the case of the tuple N -gram model, Kneser-Ney smoothing and interpolation of higher and lower N -grams are used. Extended target language models might also be obtained by adding additional information from other available monolingual corpora. However, in the translation tasks employed here, target language models are estimated by using only the information contained in the target side of the training data set.

Next, we study the impact of the N -gram model size employed in the translation system. We conduct perplexity measurements (over the development data set) obtained for N -gram models computed from the EPPS training data by using different N -gram sizes. Table 3.3 presents perplexity values obtained for translation and target language models with different N -gram sizes.

Table 3.3: *Perplexity measurements for translation and target language models of different N -gram size*

Type of model	Language	bigram	trigram	4-gram	5-gram
Translation	ES $\rightarrow$ EN	201.75	161.26	156.88	157.24
Translation	EN $\rightarrow$ ES	223.94	179.12	174.10	174.49
Language	Spanish	81.98	52.49	48.03	47.54
Language	English	78.91	50.59	46.22	45.59

The next experiment is designed to evaluate the incidence of translation and language model N -gram sizes on the overall system performance.

The full system (system **full** of previous experiment) is compared with two similar systems for which 4-grams are used for training the translation model and/or the target-language model. More specifically, the three systems compared in this experiment are:

- System **full-33**, which implements a tuple trigram translation model and a word trigram target language model. This system corresponds to the standard system configuration that was defined at the beginning of Section 3.4.
- System **full-34**, which implements a tuple trigram translation model and a word 4-gram target language model.
- System **full-44**, which implements a tuple 4-gram translation model and a word 4-gram target language model.

Table 3.4 summarizes this evaluation results for the three configurations. Again, both translation directions are considered and the optimized coefficients associated to the four feature functions are also presented for each system configuration (the log-linear weight of the translation model has been omitted from the table because its value is fixed to 1 in all cases).

Table 3.4: *Evaluation results for experiments on *N*-gram size incidence.*

Direction	System	λ_{lm}	λ_{wb}	λ_{s2t}	λ_{t2s}	mWER	BLEU
ES $\rightarrow$ EN	full-33	.49	.30	.94	.25	34.94	.5434
	full-34	.50	.54	.66	.45	34.66	.5483
	full-44	.66	.50	1.01	.57	34.59	.5464
EN $\rightarrow$ ES	full-33	.66	.73	.32	.47	40.34	.4688
	full-34	.57	.45	.51	.26	40.55	.4714
	full-44	1.24	1.07	.99	.57	40.91	.4688

As seen from table 3.4, the use of 4-grams for model computation does not provide a clear improvement in translation quality. This is more evident in the English-to-Spanish direction for which system **full-44** happens to be the worst ranked one, while system **full-33** is the one obtaining the best mWER score and system **full-34** is the one obtaining the best BLEU score. On the other hand, in the Spanish-to-English direction it seems that a little improvement with respect to system **full-33** is achieved by using 4-grams. However, it is not clear which system performs the best since system **full-34** obtains the best BLEU while system **full-44** obtains the best mWER.

According to these results, more experimentation and research is required to fully understand the interaction between the *N*-gram sizes of translation and target-language models. Notice that in the particular case of the *N*-gram SMT system described here, such an interaction is not evident at all since the *N*-gram based translation model contains by itself some of the target language model information.

3.4.3 Source-NULled Tuple Strategy Comparison

This experiment is designed to evaluate the different handling strategies for source-NULled tuples. In this section, the standard system configuration (system **full-next**) presented at the

beginning of Section 3.4, which implements the **always NEXT** strategy described in Section 3.2.1.2, is compared with a similar system (referred to as **full-lex**) implementing a more complex strategy for handling those tuples with NULL source sides using the IBM-1 lexical parameters [Bro93] for computing translation probabilities of two possible new tuple segmentations: the one resulting when the null-aligned-word is attached to the previous word, and the one resulting when it is attached to the following one (**LEX model weight** strategy). The attachment direction is selected according to the tuple with the highest translation probability. Finally, a system implementing the segmentation based on POS-entropy distributions outlined in 3.2.1.2 is also taken into account (referred to as **full-ent**).

Table 3.5 summarizes this evaluation results for systems **full-next**, **full-lex** and **full-ent**. Again, both translation directions are considered and the optimized coefficients associated to the four feature functions are also presented for each system configuration.

Table 3.5: *Evaluation results for experiments on strategies for handling source-NULLED tuples.*

Direction	System	λ_{lm}	λ_{wb}	λ_{s2t}	λ_{t2s}	mWER	BLEU
ES $\rightarrow$ EN	full-next	.49	.30	.94	.25	34.94	.5434
	full-lex	.49	.45	.78	.39	34.15	.5451
	full-ent	.55	.35	.57	.13	34.20	.5441
EN $\rightarrow$ ES	full-next	.66	.73	.32	.47	40.34	.4688
	full-lex	.96	.93	.53	.44	40.12	.4694
	full-ent	.91	.53	.73	.34	40.20	.4724

As seen from table 3.5, consistent better results are obtained in both translation tasks when using either IBM-1 lexicon probabilities or the POS entropy distribution to handle tuples with NULL source side. Even though slight improvements are achieved in both cases, specially with the English-to-Spanish translation task, results show how the initial **always NEXT** strategy is easily improved when making use of some additional knowledge.

3.4.4 Feature Function Contributions

The last experiment is designed to evaluate the relative contribution of feature functions to the overall system performance. In this section, four different systems are evaluated. These systems are:

- System **base**. It constitutes the basic N -gram translation system, which implements the tuple trigram translation model alone, *i.e.* no additional feature function is used.
- System **target-reinforced**. In this system, the translation model is used along with the target language and word bonus models.
- System **lexicon-reinforced**. In this system, the translation model is used along with the source-to-target and target-to-source lexicon models.
- System **full**. It constitutes the full system, *i.e.* the translation model is used along with all four additional feature functions. This system corresponds to the standard system configuration that was defined at the beginning of Section 3.4.

Table 3.6 summarizes this evaluation’s results, in terms of BLEU and mWER, for the four systems considered. As seen from the table both translation directions, Spanish-to-English and English-to-Spanish are considered. Table 3.6 also presents the optimized log-linear coefficients associated to the features considered in each system configuration.

Table 3.6: *Evaluation results for experiments on feature function contribution.*

Direction	System	λ_{lm}	λ_{wb}	λ_{s2t}	λ_{t2s}	mWER	BLEU
ES $\rightarrow$ EN	base	—	—	—	—	39.71	.4745
	target-reinforced	.29	.31	—	—	39.51	.4856
	lexicon-reinforced	—	—	.77	.08	35.77	.5356
	full	.49	.30	.94	.25	34.94	.5434
EN $\rightarrow$ ES	base	—	—	—	—	44.46	.4276
	target-reinforced	.33	.27	—	—	44.67	.4367
	lexicon-reinforced	—	—	.29	.15	41.69	.4482
	full	.66	.73	.32	.47	40.34	.4688

As can be observed from table 3.6, the inclusion of the four feature functions into the translation system definitively produces an important improvement in translation quality in both translation directions. Particularly, it becomes evident that features with the most impact on translation quality are the lexicon models. The target-language model and the word bonus also contribute to improve translation quality, but in less degree.

Also, although it is more evident in the English-to-Spanish direction than in the opposite one, it can be noticed from the presented results that the contribution of target-language and word bonus models is more relevant when the lexicon models are used (full system). In fact, as seen from λ_{lm} values in table 3.6, when the lexicon models are not included, the target-language model contribution to the overall translation system becomes significantly less important. A comparative analysis of the achieved translations suggests that including the lexicon models tends to favor short tuples over long ones, so the target language model becomes more important for providing target context information when the lexicon models are used.

Another important observation, which follows from comparing results between both translation directions, is that in all the cases Spanish-to-English translations are consistently and significantly better than English-to-Spanish translations. This is clearly due to the more inflected nature of Spanish vocabulary. For example the single English word “the” can generate any of the four Spanish words “el”, “la”, “los” and “las”. Similar situations occur with nouns, adjectives and verbs which may have many different forms in Spanish. This would suggest for the English-to-Spanish translation task to be more difficult than the Spanish-to-English task.

3.4.5 Error Analysis

In this section, we present a brief description of an error analysis performed to some of the outputs provided by the standard system configuration that was described in Section 3.4 (system **full**). More specifically, a detailed review of 100 translated sentences and their corresponding source sentences, in each direction, was conducted. This analysis resulted to be very useful since it allowed to identify the most common errors and problems related to our N-gram based SMT system in each translation direction.

A detailed analysis of all the reviewed translations revealed that most translation problems encountered are typically related to four basic different types of errors:

- **Verbal Forms:** A great amount of wrong verbal tenses and auxiliary forms were detected. This problem turned out to be the most common one, reflecting the difficulty of the current statistical approach to capture the linguistic phenomena that shape head verbs, auxiliary verbs and pronouns into full verbal forms in each language, especially given the inflected nature of the Spanish language.
- **Omitted Translations:** A large amount of translations involving tuples with NULL target sides were detected. Although in some cases these situations corresponded to correct translations, most of the time they resulted in omitted-word errors.
- **Reordering Problems:** The two specific situations that most commonly occurred were problems related to adjective-noun and subject-verb structures.
- **Agreement Problems:** Inconsistencies related to gender and number were the most commonly found.

Table 3.7 presents the relative number of occurrences for each of the four types of errors identified in both translation directions.

Table 3.7: *Percentage of occurrence for each type of error in English-to-Spanish and Spanish-to-English translations that were studied*

Type of Error	English-to-Spanish	Spanish-to-English
Verbal Forms	31.3%	29.9%
Omitted Translations	22.0%	26.1%
Reordering Problems	15.9%	19.7%
Agreement Problems	10.8%	4.6%
Other Errors	20.0%	19.7%

Notice from table 3.7 that the most common errors in both translation directions are those related to verbal forms. However, it is important to mention that 29.5% of verbal-form errors in the English-to-Spanish direction actually correspond to verbal omissions. Similarly, 12.8% of verbal-form errors in the Spanish-to-English direction are verbal-omissions. According to this, if errors due to omitted translations and to omitted verbal forms are considered together, it is evident that errors involving omissions constitute the most important group, specially in the case of English-to-Spanish translations. It is also interesting to notice that the Spanish-to-English direction exhibits more omitted-translation errors that are not related to verbal forms than the English-to-Spanish direction.

Also from table 3.7, it can be noticed that agreement errors affect more than twice English-to-Spanish translations than Spanish-to-English ones. This result can be explained by the more inflected nature of Spanish.

Finally, as an illustrative example, three Spanish-to-English translation outputs are presented below. For each presented example, errors have been boldfaced and correct translations are provided in brackets:

- *The policy of the European Union on Cuba* **NULL must** *[must not] change* .
- *To achieve these purposes , it is necessary* **NULL** *for the governments* **to be allocated** *[to allocate] , at least , 60 000 million* **NULL** *dollars a year ...*
- *In the UK we have* **NULL** *[already]* **laws enough** *[enough laws] , but we want to encourage* **NULL** *other States ...*

3.5 Contrasting Phrase-based SMT

In this section we focus on the singularities of the N -gram-based system when compared to a standard phrase-based system. First, we point at the translation units employed in both approaches and the underlying translation models. Finally, we carry out a performance comparison of both approaches under different training data constraints.

3.5.1 Phrase-based Translation Model

Both translation Models are founded on bilingual units, *i.e.* two monolingual fragments where each one is supposed to be the translation of its counterpart. They actually consists of the core of the translation systems. In the bibliography, phrase-based units are typically referred to as **phrases** (while **tuples** is typically used for the N -gram-based translation units).

Section 3.2.1.1 details the extraction of tuples in the N -gram-based approach. Regarding phrases, the extraction employs also word-to-word alignments of the training corpus. A standard definition of phrases considers the set of units described as **any** pair of source and target words that satisfies the next two basic constraints [Och04b]:

- Words are consecutive along both sides of the bilingual phrase, and
- no word on either side of the phrase is aligned to a word out of the phrase.

Figure 3.7 illustrates the process of phrases (right) and tuples (bottom) extraction from a given pair of word-to-word aligned sentences.

The first singularity regarding both translation units considers the extraction methods. As it can be seen in figure 3.7, whereas the sentence pair can be segmented into multiple phrase sets ($[p1 + p9 + p12 + p15]$, $[p2 + p13]$, $[p3 + p15]$, *etc.*), only one segmentation is possible when extracting tuples ($[t1 + t2 + t3 + t4]$).

This multiple segmentation, employed in the extraction of phrases, turns the phrase-based approach more robust to noisy alignments than the N -gram-based approach. An erroneous alignment introduced in a sentence pair forces (typically) the apparition of a long tuple that hides the information of the internal links. Figure 3.8 illustrates this situation. The erroneous alignment ' $I \rightarrow un$ ' forces the apparition of the long tuple ' $I.must.buy.a\#debo.comprar.un$ ', losing the translation information of the alignments present in the tuple. On the other hand, the set of phrases is also affected by the introduction of the wrong alignment, phrases $p1$, $p2$, $p8$ and $p9$ of the original alignment in Figure 3.7 are lost. However, all phrases which do not take the wrong alignment into account survive in the new phrase set.

Additionally, the use of long tuples impoverishes the probability estimates of the translation model, as longer tuples appear less often in training than the smaller ones (data sparseness problem). Therefore, language pairs with important differences in word order may suffer from poor probability estimates.

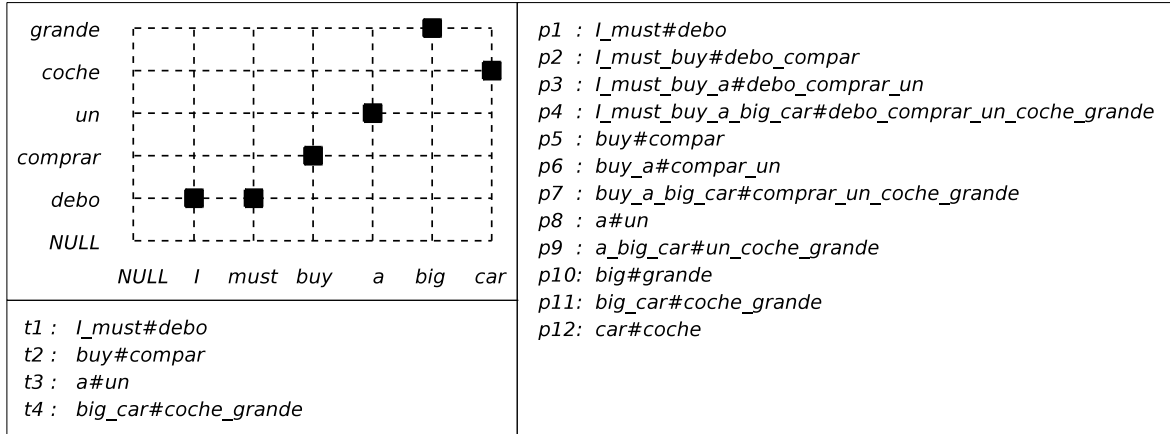


Figure 3.7: *Phrase and tuple extraction.*

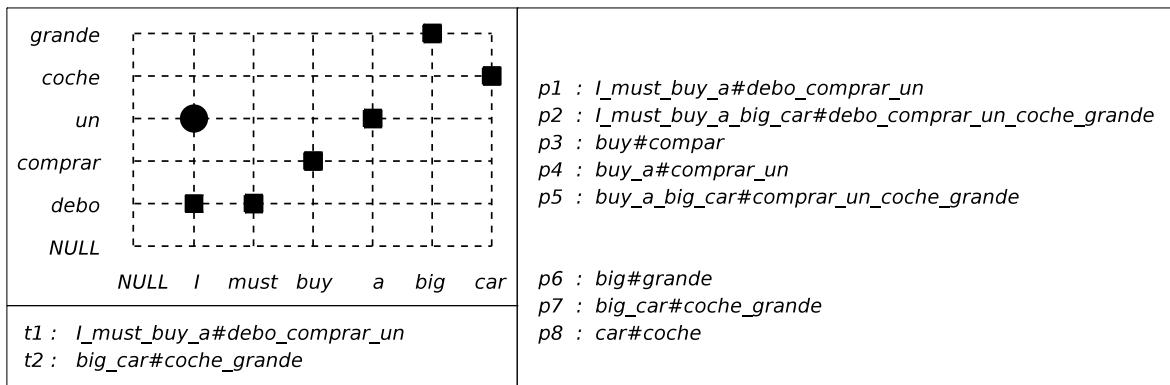


Figure 3.8: *Phrase and tuple extraction with noisy alignments.*

Following the unit extraction methods of both approaches, tuples consist of a subset of the set of phrases, with the exception of those tuples with the '*NULL*' word in their target side which can not appear as phrases. However, these (target-NULLed) tuples are not designed to be used uncontextualized, but as part of a sequence of tuples (tuple *N*-gram), for which the equivalent phrase must also exist. Therefore, we can consider the set of tuples as a strict subset of the set of phrases.

From the previous observation, we can derive that the generation power of the phrase-based systems is higher (or at least the same) to that of the N -gram-based systems. Here, we use 'generation power' to account for the number of different translation options that can be hypothesized by means of the translation units available in a translation system. This fact is specially relevant for language pairs with strong differences in word order, where long tuples

appear more often (also noisy alignments) boosting the consequent information loss (hidden links within long tuples) and reducing the generation power.

An example of this situation is shown in section 4.3, where we introduce a simple strategy to clean noisy alignments making use of linguistic information (shallow parsing). Results show that the strategy is specially relevant for the N -gram-based approach and for some language pairs with strong reordering needs (Arabic-English), while no effect is appreciated on the phrase-based approach and for other language pairs with less reordering needs (Spanish-English).

Notwithstanding the fact that the two approaches rely on different translation models, both follow the same generative process. It is composed of two main steps:

- **Source unit segmentation**, the reordered input sentence is segmented into sequences of source words which are to be translated jointly.
- **Translation choice**, where each sequence of source words selects the target side to which it is linked to.

Figure 3.9 shows that both approaches follow the same generative process, differing on the structure of translation units. Whereas the phrase-based approach employs translation units uncontextualized, the N -gram-based approach takes the translation unit context into account. In the example, the units ' $s3\#t1$ ' and ' $s1_s2\#t2_t3$ ' of the N -gram-based approach are used considering that both appear sequentially. This fact can be understood as using a longer unit that includes both (longer units are drawn in grey).

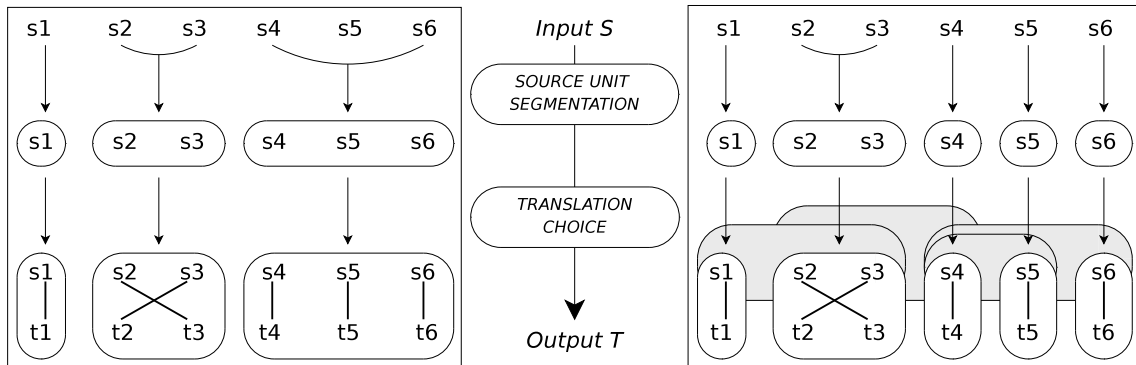


Figure 3.9: Generative process. Phrase-based (left) and N -gram-based (right) approaches.

In consequence, the translation context is introduced in phrases by the use of word sequences in both sides of the translation unit, while tuples model the context within the tuple unit (as phrases) and by taking the sequence of units into account.

Another important difference between phrases and tuples is that the former do not need to take a hard decision to avoid source-NULLED units (see Section 3.2.1.2). The extraction algorithm ensures that phrases with source or target NULL side never appear.

Additionally, the embedded word units (see Section 3.2.1.3) used in the N -gram-based approach are not needed in the phrase-based approach. As already stated, the N -gram-based

approach suffers highly from the apparition of longer units. The internal links of a long tuple are discarded, causing the waste of a lot of translation information. The multiple segmentation of phrases tends to alleviate this problem producing the longer as well as the shorter units (see Figure 3.7).

Phrase translation probabilities are typically estimated by relative frequency,

$$p(\tilde{s}|\tilde{t}) = \frac{N(\tilde{s}, \tilde{t})}{N(\tilde{t})} \quad (3.9)$$

where $N(\tilde{s}, \tilde{t})$ consists of the number of times the phrase $\tilde{s}$ is translated by $\tilde{t}$. In order to reduce the overestimation problem derived of sparse data (*the maximum probability, $p = 1$, is assigned to a phrase with target side occurring only once in training*), the posterior phrase conditional probability is also taken into account, $p(\tilde{t}|\tilde{s})$.

Both translation model probabilities can be introduced in a phrase-based system like a log-linearly combined feature function. As described by the following equation:

$$p_{RF_{s2t}}(s_1^J, t_1^I) \approx \prod_{k=1}^K p(s|t)_k \quad (3.10)$$

where $p(s|t)_k$ is the phrase translation probability of the k -th phrase in the overall search (written as $p(\tilde{s}|\tilde{t})$ in Equation 3.9). $p_{RF_{s2t}}$ is analogously defined for the opposite translation direction.

3.5.2 Translation Accuracy Under Different Data Size Conditions

The next experiments are conducted in order to test the ability of both approaches to adapt to different training conditions in terms of data availability.

In order to make a fair comparison, we have built two systems which share the most of their components (training corpus, word-alignments, decoder, models *etc.*). They obviously diverge on the translation model and some of the additional models used as feature functions. Table 3.8 shows a summary of the models used by each system. The phrase bonus model (*PB*) and the relative frequency models computed for both directions (RF_{s2t} and RF_{t2s}) are only used by the phrase-based system (*pbsmt*), while only the N -gram-based system (*nbsmt*) employs the N -gram translation model (*BM*). Target language model (*LM*) word bonus (*WB*) and translation lexicon models (LEX_{s2t} and LEX_{t2s}) are shared by both systems.

Table 3.8: *Models used by each system.*

System	<i>BM</i>	RF_{s2t}	RF_{t2s}	<i>LM</i>	<i>WB</i>	<i>PB</i>	LEX_{s2t}	LEX_{t2s}
<i>nbsmt</i>	1	0	0	1	1	0	1	1
<i>pbsmt</i>	0	1	1	1	1	1	1	1

Three different training data size conditions are considered: *full*, *medium* and *small* (detailed in section A.1.2). Results are shown in Table 3.9.

As it can be seen in both translation directions, the *N*-gram-based system (slightly) outperforms the phrase-based system under small data conditions.

Table 3.9: Accuracy results under different training data size conditions.

Direction	Training size	<i>nbsmt</i>		<i>pbsmt</i>	
		mWER	BLEU	mWER	BLEU
ES $\rightarrow$ EN	<i>full</i>	34.49	55.07	34.39	55.43
	<i>medium</i>	37.17	51.26	37.12	51.15
	<i>small</i>	44.26	40.94	44.28	40.21
EN $\rightarrow$ ES	<i>full</i>	41.59	48.06	40.75	47.73
	<i>medium</i>	45.02	43.10	45.16	43.46
	<i>small</i>	53.18	31.89	53.71	31.28

These results are somehow unexpected. In principle, the phrase-based approach seems to make a better profit of the training data because of considering multiple segmentations of the source/target words in the phrase extraction process, in contrast to the single best segmentation of the tuple extraction process. Hence, this condition should be specially significant under scarce data availability.

Nevertheless, accuracy results do not support the previous hypothesis. Under large data size conditions (**full** and **medium**) both approaches do not show important differences in performance. However, when it comes to a **small** data size condition, accuracy results seem to favor the election of an *N*-gram-based system, which obtain better results in both translation directions and for both evaluation scores.

A similar behavior is observed when considering out-of-domain test conditions. Results obtained by the UPC *N*-gram-based system in different translation evaluations show how the system is better ranked (compared to other phrase-based systems) under out-of-domain conditions. In Appendix B are detailed the participation of the system in several international translation evaluations. Our system was highly ranked in the TC-Star (B.1), IWSLT (B.2 and WMT (B.3) tasks when considering out-of-domain conditions than under in-domain conditions.

A more detailed study on these unexpected behaviors needs to be conducted for a better understanding of the difference in performance.

3.6 Chapter Summary and Conclusions

This chapter introduced in detail *N*-gram-based SMT. It starts with the definition of the bilingual *N*-gram translation model and examines the contribution of each feature function. The system is founded on the maximum-entropy approach, implemented as a log-linear combination of different feature models.

We reported in depth the singularities of the core translation model employed in the SMT system. Details are given on the translation unit extraction and refinement, *N*-gram modeling, contribution of additional models, system architecture and tuning. Accuracy results are presented on a large-sized Spanish-English translation task, showing the contribution in translation accuracy of each system component.

A manual error analysis has also been carried out to further study the output of the translation system. It revealed the most common errors produced by the system, categorized in four main groups: verbal forms; omitted translations; word ordering and agreement problems.

Finally, the presented N -gram-based SMT system is contrasted to a standard phrase-based system. Singularities of both approaches are outlined, which arise motivated by the idiosyncrasy of either translation units employed. A performance comparison is also conducted under different training data-size constraints.

Notice, that the work presented in this chapter has been jointly carried out with the rest of members of the UPC SMT group. The system described in this chapter has been presented to several evaluation campaigns attaining state-of-the-art results under monotonic translation tasks. Evaluation campaigns are detailed in Appendix B.

Chapter 4

Linguistically-motivated Reordering Framework

This chapter describes an elegant and efficient approach to couple reordering and decoding. The reordering search problem is tackled through a set of linguistically motivated rewrite rules, which are used to extend a monotonic search graph with reordering hypotheses. The extended graph is traversed during the global search when a fully-informed decision can be taken. Different linguistic information sources are considered, employed to learn valid permutations under the reordering framework introduced. Additionally, a refinement technique of word alignments is presented which employs shallow syntax information to reduce the set of noisy alignments present in an Arabic-English task. The chapter is organized as follows:

- Firstly, in §4.1.2 we review the basic features of the N -gram-based system presented in the previous chapter focusing on the lack of reordering abilities, what motivates its extension with the reordering framework presented in this chapter. Note that the reordering framework here introduced can also be applied on a SMT system build following the phrase-based approach.
- The reordering framework is presented in §4.2. We give details of how reordering rules are automatically extracted from word alignments using different linguistic information sources. We analyze the models used to help the decoder make the right reordering decision, and finally, we also give details of the extension procedure of the monotonic path into a permutations graph.
- §4.3 reports the experiments conducted in order to evaluate the impact in translation quality of the presented approach. Experiments are carried out on different data size tasks and language pairs: Spanish-English, Arabic-English and Chinese-English. In this section we propose a word alignment refinement technique which reduces the set of noisy alignments of the Arabic-English task.
- In §4.4 a summary of the chapter can be found, highlighting the main conclusions extracted from it.

4.1 Introduction

As introduced in §2 the first SMT systems worked at the word level [Bro90]. In this first systems, differences in word order between source and target languages made reordering a very hard problem in terms of both modeling and decoding. In [Kni99], the search problem is classified NP-complete when arbitrary word reorderings are permitted, while polynomial time search algorithms can be obtained under monotonic conditions.

The apparition of phrase-based translation models brought a clear improvement in the state-of-the-art of SMT [Zen02]. The phrase-based approach introduced bilingual phrases (contiguous sequences of words in both languages) as translation units which naturally capture local reorderings, thus alleviating the reordering problem. However, the phrase-based approach did not entirely solve the reordering problem, showing a main weakness on longest reorderings. Large-distance reorderings need long phrases, which are not always present in the training corpus because of the obvious data sparseness problem.

In recent years huge research efforts have been conducted aiming at developing improved reordering approaches. In the next section several of the proposed alternatives are discussed.

4.1.1 Related Work

As we have previously outlined, the first SMT systems introducing reordering capabilities were founded on the brute force of computers. They intended to find the best reordering hypothesis through traversing a fully reordered search graph, where all permutations of source-side words were allowed. This approach resulted computationally very expensive for even very short input sentences. Hence, in order to make the search feasible, several reordering constraints were developed: under **IBM** constraints, each new target word must be aligned to one of the first k uncovered source words [Bro93]; under **Local** constraints, a given source word is only allowed to be reordered k positions far from its original position [Kan05]; the **MaxJumps** constraint limits the number of reorderings for a search path (whole translation) to a given number [Cre05c]; and finally, **ITG** [Wu96] constraints, where the input sentence is seen as a sequence of blocks, and pair of blocks are merged by either keeping the monotonic order (original) or inverting their order. This constraint is founded on the parse trees of the simple grammar in [Wu97]. The use of these constraints implied a necessary balance between translation accuracy and efficiency.

Additionally to the previous search constraints, a distance-based reordering model is typically used during the search to penalize longest reorderings, only allowed when well supported by the rest of models. More recently, lexicalized reordering models have been introduced, which score reorderings in search using distance between words seen in training [Til04, Kum05], distance between phrase pairs [Til05, Nag06], based on adjacency/swap of phrases [Col05b], and using POS tags, lemmas and word classes to gain generalization power [Zen06].

A main criticism to this brute force approach is the little use of linguistic information to limit the reorderings needed, while in linguistic theory, reorderings between linguistic phrases are well described.

Current (phrase-based) SMT systems tend to introduce linguistic information into new reordering strategies to overcome the efficiency problem. Several alternatives have been proposed:

- Some approaches employ deterministic reordering, where a preprocessing step is performed aiming at transforming the order of the source sentence to make it closer to the target language.

[Ber96] describes a reordering approach for a French-English task that swaps sequences like ‘*noun₁ de noun₂*’.

[Col05b, Wan07] employ manually created reordering rules based on syntax information for Chinese-English translation. In [Xia04, Hab07] rules are automatically extracted from the training corpus making use of word alignments for Chinese-English and Arabic-English SMT. [Nie04] describes an approach for German-English translation that combines verbs with associated particles, and reorders questions too. [Pop06c] uses POS information to automatically learn reorderings for a Spanish-English task. [AO06] outlines the main weakness of this approach, consisting of the deterministic reordering choice, which is taken separately from the overall search.

- The same language word order harmonization idea is followed in works such as [Cre06a, Cj06, Zha07, Cre07b], differing on the fact that the reordering decision is taken fully coupled with the SMT decoder by means of an input graph which provides the decoder with multiple reordering options. Hence, a fully-informed reordering decision is taken in consensus by the whole SMT models.

Note that none of the previous approaches employ syntax directly to the decoding step. In contrast, [Chi05] makes use of a synchronous context-free grammar introducing a hierarchical approach to reordering (no parsing is required). [Din05, Qui05, Liu06, Hua06] make use of syntax information of the source language in a transducer-style approach. [Yam01, Mar06] build a full parse tree in the target language, allowing hierarchical reordering based on synchronous grammars.

4.1.2 *N*-gram-based Approach to SMT

In Chapter 3, we have seen that given a word alignment, tuples define a unique and monotonic segmentation of each bilingual sentence, allowing *N*-gram estimation to account for the history of the translation process. Therefore, like under the phrase-based approach, the word context is introduced in translation units (tuples). Additionally, it relies on the sequence of tuples, bilingual *N*-gram, to account for larger sequences of words which, similarly to the phrase-based approach, alleviates the reordering problem. However, the structure of tuples poses important problems to the *N*-gram-based system under language pairs with important structural disparities.

In §3.5 we outlined the singularities of phrases and tuples and between their respective models. Disparities between source and target training sentences forces the apparition of large tuples, which imply an important loss of information contained in the internal *hidden* links. Additionally, the sequential structure of the *N*-gram language model further hurts the system in contrast to the phrase-based approach. The external context of a reordered unit typically reinforces the monotonic hypotheses of new sentences, opposing to reordered ones. Figure 4.1 illustrates this situation. A word-to-word alignment (top left) is used to extract a set of tuples (top right) following the procedure described in §3.2.1.1. It is also shown a permutation graph computed for a test sentence to be translated.

Notice that the source training sentence (top) and the test sentence (bottom) differ on a single word (*flight/trip*).

As it can be seen, additionally to the sparseness problem of translation units when reordering appears (very large units), the N -gram model tends to higher score monotonic hypotheses. The reason is that monotonic sequences, like '*does the*' and '*last today*', are more likely to have been seen in training than the corresponding of the reordered path ('*does last*' and '*flight today*') reinforcing the monotonic path of the search. Monotonic sequences are more likely to exist in training because they contain the source words in the original order, which is the order employed when estimating the N -gram language model of tuples.

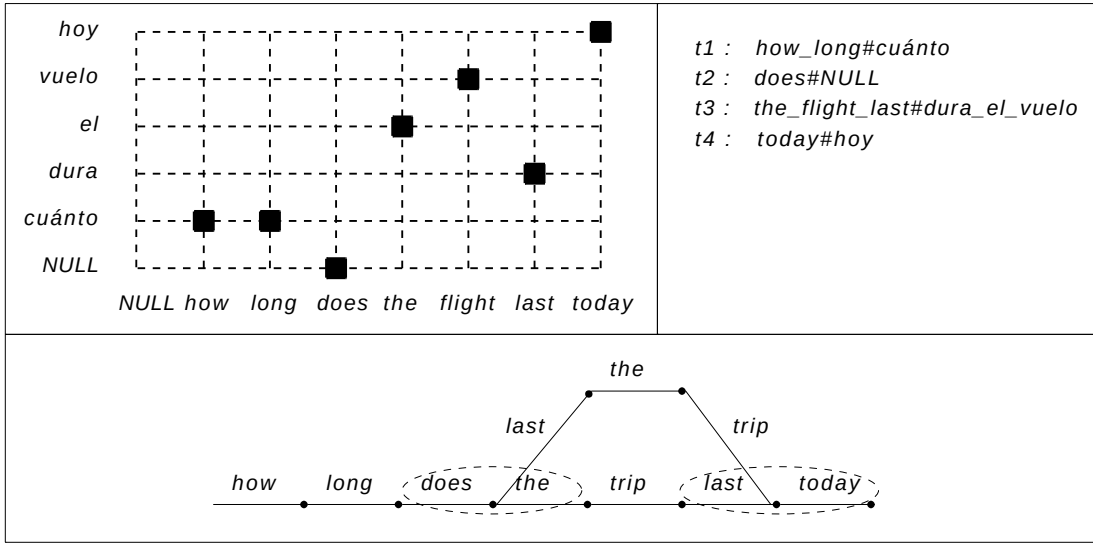


Figure 4.1: Tuples (top right) extracted from a given word aligned sentence pair (top left) and permutation graph (bottom) of the input sentence: 'how long does the trip last today'.

It is worth noticing that this situation is only relevant when the tuple '*the trip last#dura el vuelo*' does not exist in training. Otherwise, the decoder would probably use it, following again the monotonic path.

4.2 Reordering Framework

Now, we introduce the reordering framework presented in this chapter. It is composed of a double-sided process.

In training time, a set of reordering rules are automatically learned, following the word-to-word alignments. Source-side words are reordered aiming at monotonizing the source and target word order. For each distortion introduced in training, a record in the form of a reordering rule is taken.

Later in decoding time, the set of rules are employed to build a permutation graph for each input sentence, which provides the decoder with a set of reordering hypotheses.

Figure 4.2 illustrates the generative translation process followed by our system when in-

roduced the reordering framework. As it can be seen, it contrasts to the generative process presented in Figure 3.9 by introducing reordering over the source words in the first step.

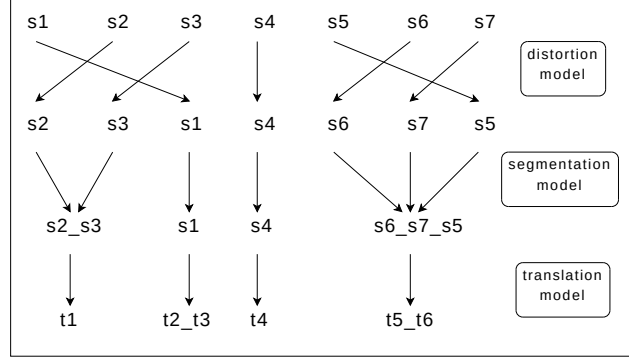


Figure 4.2: Generative translation process when introducing the reordering framework.

Under this approach, translation units are extracted considering the (source) reordered corpus. Consequently, reorderings do not only help in decoding (providing reordering hypotheses) but also by extracting less sparse translation units. In [Kan05, Cj06, Col05b] is suggested a similar procedure previous to the phrase extraction.

4.2.1 Unfold Tuples / Reordering Rules

As introduced in the previous lines, the extraction of translation units and reordering rules are performed tightly coupled.

Coming back to the definition of tuples in §3.2.1.1, a tuple can be seen as the minimum sequence of source and target words which are not word-aligned out of the tuple. From another point of view, each discrepancy in the word order between source and target words (reordering) is captured within a tuple.

The latter point of view fits exactly with the property of the rules that we are now interested on. A reordering rule identifies the sequences of source words for which the corresponding target words follow a different order. Additionally, the rule indicates the distortion needed on the source words to acquire the order of the target words.

More formally, a reordering rule consists of the rewrite pattern $s_1, \dots, s_n \rightarrow i_1, \dots, i_n$, where the left-hand side $s_1, \dots, s_n$ is a sequence of source words, and the right-hand side $i_1, \dots, i_n$ is the sequence of positions into which the source words are to be reordered.

Figure 4.3 (bottom) shows an example of reordering rule that can be read as: a source sentence containing the sequence 'the flight last' is to be reordered into 'last the flight'. Reordering is encoded in the right-hand side of the rule using the relative positions of the words in the sequence '2:last 0:the 1:flight').

To extract rewrite patterns from the training corpus we use the crossed links found in translation tuples. A rewrite pattern can also be seen as the reordering rule that applied over the source words of a tuple generates the word order of the tuple target words. Figure 4.3 illustrates

the extraction of rewrite patterns. It can be seen a translation tuple with its internal word alignments (top left). The figure also shows a set of three tuples, hereinafter referred to as **unfold** tuples, extracted following the new technique (detailed next) by '*unfolding*' the original tuple, also referred to as **regular** tuple. As it can be seen, the word alignment is monotonized when the pattern is applied over the source words of the *regular* tuple.

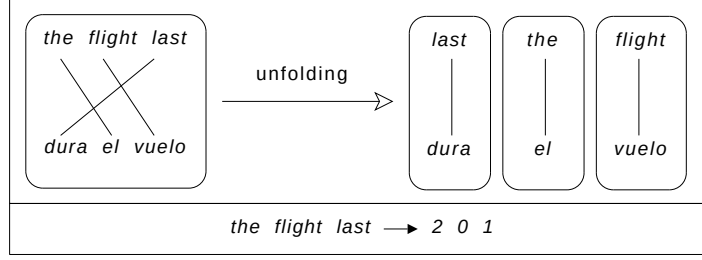


Figure 4.3: Pattern extraction.

Additionally, each pattern is scored with a probability computed on the basis of relative frequency:

$$p(s_1, \dots, s_n \rightarrow i_1, \dots, i_n) = \frac{N(s_1, \dots, s_n \rightarrow i_1, \dots, i_n)}{N(s_1, \dots, s_n)} \quad (4.1)$$

So far, we have defined the source side of the reordering rules as the sequence of source words contained in the original *regular* tuples. Considering the target side, it consists of the positions of the same source words after being distorted by means of the unfolding technique detailed next. Hence, reordering rules and *unfold* tuples are tightly coupled.

Figure 4.4 shows the unfolding procedure applied over three different alignment structures considering the nature of the alignments (one-to-one, one-to-many, many-to-one). The unfolding technique makes use of the word alignments. It can be decomposed in three main steps:

- First, words of the target side are grouped when linked to the same word in the source side. When grouping two target words (*i.e.* 'X' and 'Z' of tuple *c*), all words between them (in this case the target word 'Y') are also introduced in the new group. The group inherits the links of the words it is composed of (*i.e.* 'XYZ' inherits the links of 'X', 'Y' and 'Z').
- In the second step, new groups between source and target words (or groups) are formed when connected through an alignment. Groups are marked in the figure with dotted circles.
- Finally, the resulting groups become unfold tuples and are output following the original order of the target words of each unit.

Considering the unfold technique, we can conclude that regular tuples containing crossings produced by 1-to-1 alignments are easily unfold, ending up in very small (less sparse) and reusable units (*i.e.* tuple *a*). In regard of those regular tuples containing crossings where 1-to-*N* alignments (for $N > 1$) are implied, when the *N* is referred to the source words (*i.e.* tuple *b*), the unfolding is successfully applied, resulting also in smaller units. However, when the *N* is

referred to the target words (*i.e.* tuple c) the regular tuple can not be unfold. Theoretically, the same unfolding could be applied to the latter units moving accordingly the target words, producing the sequence of unfold units: ' $A\#Y B\#X_Z$ '. However, notice that using the units of this sequence, the valid target sentence ' $X Y Z$ ' can not be hypothesized since source-side reorderings are only used.

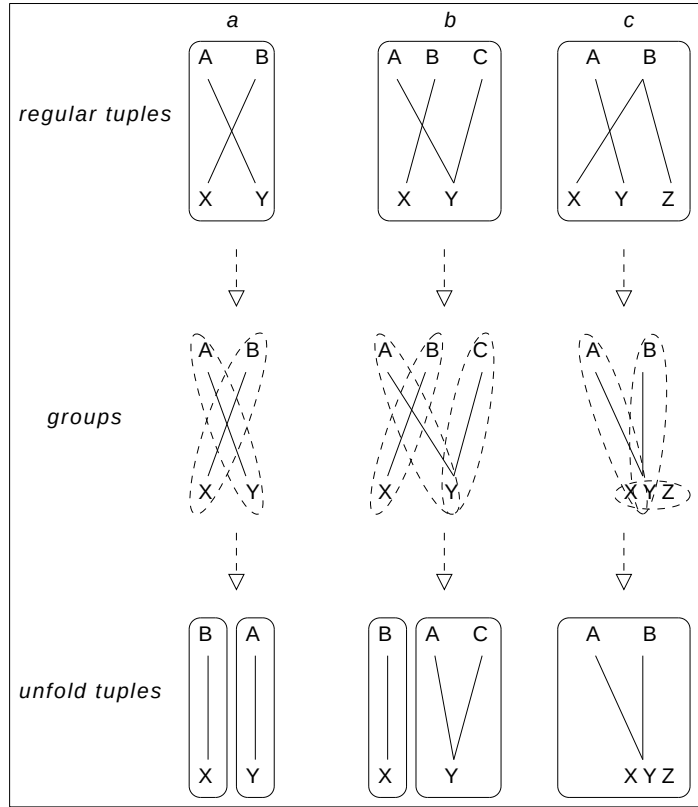


Figure 4.4: Tuple extraction following the unfold technique.

An important *weakness* of this reordering framework arises motivated by the generation process followed by our translation system, where only source-side reorderings are available. Figure 4.5 clearly illustrates this situation. The source word '*ocurrió*' is aligned to two very distant target words '*did*' and '*happened*' which prevent the tuple being unfolded (left). The example employs the word '*...NP...*' to account for an arbitrary large noun phrase.

Further research work must be conducted in order to reduce the sparseness problem derived of crossings with 1-to- N alignments. The sequence of units shown in Figure 4.5 (right) outlines an envisaged solution. It introduces an additional source word '*[did]*' which breaks the 1-to- N alignment into N 1-to-1 alignments. Hence, allowing to apply the unfolds previously detailed.

This solution introduces new words into the input sentence. We leave this solution as further work, where we plan to tackle the problem by means of an input graph with paths considering different number of input words.

Summing up, we have seen that reordering rules and unfold units are tightly coupled techniques. The introduced reordering framework aims at reducing the sparseness problem of long

units by unfolding the internal crossings of word-alignments. As a consequence, when translating new sentences the approach needs some distortion of the source words to acquire the right order of the target sentence. Distortion is introduced in the form of reordering rules, which are learnt in training from the same unfolds used to monotonize the word order.

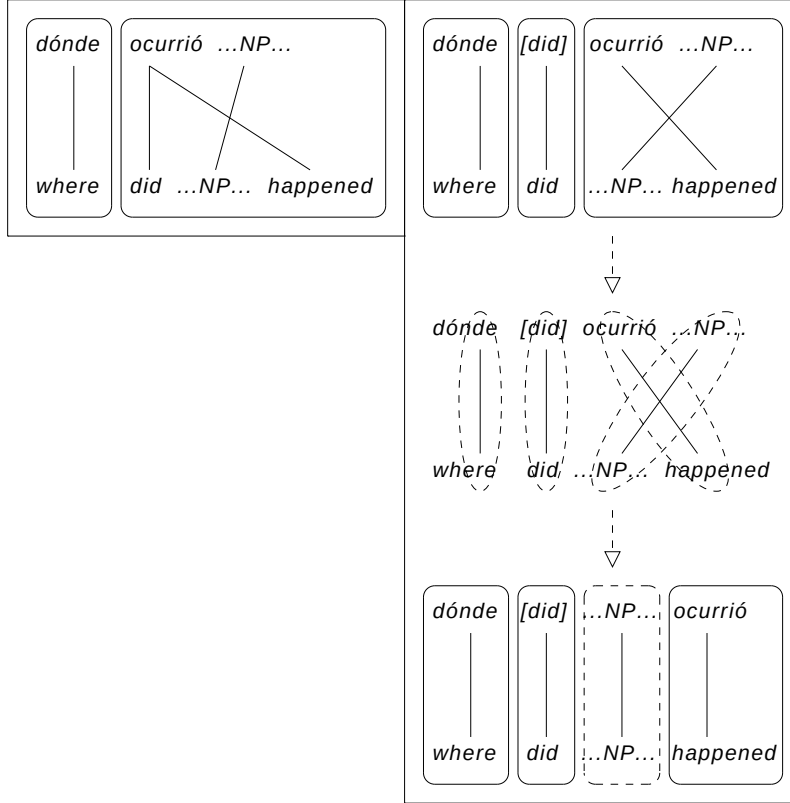


Figure 4.5: 1-to-N alignments can not be unfold (left). Envisaged solution (right).

In Equation 4.1 we have introduced a probability computed over each reordering rule. Given that reordering rules are extracted from word alignments (which are computed automatically), the apparition of noisy alignments introduces also noisy rules. This is, rules which are not motivated by disparities in the word order of the source and target sentences but by erroneous alignments. In order to filter out some of these rules we employ the probability of Equation 4.1 to prune out all but the rules which achieve a given threshold (empirically set).

Additionally, in contrast to the model built from regular units (see Figure 4.1) the new N -gram translation model estimated with unfold units reinforces reordered hypotheses as they contain reorderings more likely to have been seen in training (*i.e.* sequences 'does last' and 'trip today' of Figure 4.6). These sequences and the training source sentences have been similarly reordered.

Somehow the N -gram translation model is also acting as a reordering model. It scores differently hypotheses which contain the same target words but differently ordered. Giving a higher score to those reordering hypotheses which follow a reordering introduced also for the training data.

into account: generalization power, accuracy and sparseness of the resulting rules. Additionally, connected with the accuracy of rules, the accuracy of the processes which automatically compute the linguistic information (tagging, chunking, parsing) also needs to be considered.

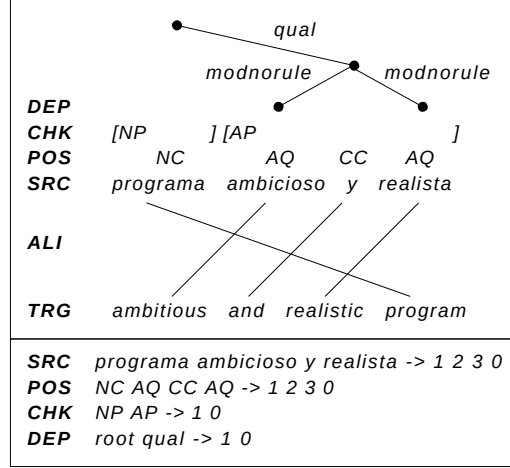


Figure 4.7: Linguistic information used in reordering rules.

While sparseness is always related to the amount of data used to collect reliable statistics, needing for being empirically adjusted, generalization power and accuracy of the rules can be seen as two sides of the same problem. Generalization power alludes to the ability of rules to capture unseen events. For instance, if we are translating from English to Spanish, the rule '*white house* $\rightarrow$ 1 0' has a very little generalization power. It can only be applied on the event '*white house*'. On the other hand, considering accuracy, the rule has a high level of accuracy. It results difficult to imagine an example where the sequence '*white house*' is not translated into Spanish as '*casa blanca*'.

If instead of the previous rule we employ '*JJ NN* $\rightarrow$ 1 0', where *JJ* and *NN* stand respectively for *adjective* and *noun*, the new rule has gained in generalization power. All sequences composed of '*adjective* + *noun*' are now captured (*i.e.* '*blue house*', '*yellow house*', '*white table*', *etc.*). However, the accuracy is reduced when compared to the initial rule. It is not difficult to imagine examples of English sequences of '*adjective* + *noun*' which are not swapped when translated into Spanish (*i.e.* '*great idea* $\rightarrow$ *gran idea*', '*good year* $\rightarrow$ *buen año*', *etc.*). Hence, when building reordering rules, we need to balance their accuracy and generalization power (inversely related features).

Furthermore, since the previous rule can be simply stated as '*noun adjective* $\rightarrow$ 1 0' the system needs a (potentially) infinite number of rules, sequences of POS tags indicating a noun phrase followed by an adjective phrase, to capture all the possible examples (recursive feature of natural languages). In other words, the generalization power of POS-based rules is somehow limited to short rules (less sparse) which fail to capture many real examples. Longest rules typically respond to reorderings between full (linguistic) phrases, which are not restricted to any size. In order to capture this long-distance reorderings we introduce rules with tags referred to arbitrary large sequences of words (chunks or syntax subtrees).

The framework proposed in this chapter does not aim at performing hard reordering decisions

(which need to be highly accurate) but to couple reordering and decoding. This is, our concern at this point is to introduce a set of reordering hypotheses into the global search which hopefully contains the successful one/s. The final decision is delayed to the global search, where all models are available.

Our main objective is to select the minimum (for efficiency reasons) set of reordering hypotheses containing the right one/s. Hence, the stress is put on the generalization power. We need rules able to capture the most of the unseen events, at the minimum computational cost.

Additionally, one of the initial difficulties we face when introducing linguistic information in the translation process is the apparition of noisy data. As any other technique based on machine learning, the ideal condition of using clean (exact) data can not be assumed. Furthermore, the multiple processes (and language tasks) employed to extract linguistic information have very different accuracy levels, which must be understood as an additional variable of the translation process. Typically, POS tagging is known to obtain higher accuracy rates than chunking, which also achieves usually better results than parsing.

POS-tags

Part-of-speech tagging, also called grammatical tagging, is the process of marking up the words in a text as corresponding to a particular part-of-speech (lexical category, word class or lexical class), based on its definition, as well as on its context, *i.e.* relationship with adjacent and related words in a phrase, sentence, or paragraph. POS tagging in the context of computational linguistics, employs algorithms which associate discrete terms, as well as hidden parts of speech, in accordance with a set of descriptive tags.

A part-of-speech is a linguistic category of words (or more precisely lexical items), which is generally defined by the syntactic or morphological behavior of the lexical item in question. Examples of part-of-speech are: *adjectives, adverbs, nouns, verbs, clitics, conjunctions, determiners (articles, quantifiers, demonstrative and, possessive adjectives), pronouns, etc.*

The accuracy reported for POS tagging systems is higher than the typical accuracy of very sophisticated algorithms that integrate part of speech choice with many higher levels of linguistic analysis: syntax, morphology, semantics, and so on (discussed later).

Some POS taggers work producing a tag set which includes additional information, such as: gender, number, time (of verb), type (of adjective, pronoun, *etc.*), *etc.*

For instance, the sentence: *'the boy looks at the man with the telescope'* can be POS tagged as follows:

[DT the] [NN boy] [VBZ looks] [IN at] [DT the] [NN man] [IN with] [DT the]
[NN telescope]

where, tags '*NN, VBZ, DT IN*' stand respectively for *noun, verb, determiner* and *preposition*.

Considering reordering rules using POS tags, they are very similarly defined to the rules using raw words. The left-hand side of the rule consists of a sequence of POS tags. This sequence is referred to the source side of a regular tuple that is to be reordered by means of the unfolding technique (previously detailed). The right-hand side consist of the positions of the same POS tags (right-hand side) after being distorted by means of the unfolding technique.

The probability computed for each pattern (shown in Equation 4.1) is now employed replacing the sequence of source words $s_1, \dots, s_n$ by the sequence of POS tags $p_1, \dots, p_n$.

Considering a given training corpus and word alignment, the same number of reordering rules are extracted when employing source words or POS tags. However, the vocabulary of rules extracted using raw words is typically much higher than that of the rules using POS tags, which indicates that apart from having a higher generalization power, rules from POS tags are much less sparse.

Chunks

Chunking (also shallow parsing or 'light parsing') is an analysis of a sentence which identifies the constituents (noun groups, verbs,...), but does not specify their internal structure, nor their role in the main sentence. Text chunking is an intermediate step towards full parsing and is founded on a previous POS tagging analysis of the sentence to be chunked.

The previous sentence: '*the boy looks at the man with the telescope*' can be chunked as follows:

[NP the boy] [VP looks] [PP at] [NP the man] [PP with] [NP the telescope]

where, phrase tags '*NP, VP, PP*' stand respectively for *noun phrase, verbal phrase* and *prepositional phrase*.

Mainly, chunk-based rules allow the introduction of phrase tags in the left-hand side of the rules. For instance, the rule: '*VP NP* $\rightarrow$ 1 0' indicates that a verbal phrase '*VP*' preceding a noun phrase '*NP*' are to be swapped. This is, the sequence of words composing the verbal phrase are reordered at the end of the sequence of words composing the noun phrase.

In training, like POS-based rules, a record is taken in the form of a rule whenever a source reordering is introduced by the **unfold** technique. To account for chunk-based rules, a phrase tag is used instead of the corresponding POS tags when the words composing the phrase remain consecutive (not necessarily in the same order) after reordered. Notice that rules are built using POS tags as well as phrase tags. Since both approaches are founded on the same reorderings introduced in training, both (POS- and chunk-based rules) collect the same number of training rule instances.

Figure 4.8 illustrates the process of POS- and chunk-based rule extraction. Word-alignments, chunk and POS information (top), *regular* and *unfold* translation units (middle) and reordering rules (bottom) are shown.

In the previous example, the reordering rule is applied over the sequence ' $s_2 s_3 s_4 s_5 s_6$ ', which is to be transformed into ' $s_6 s_5 s_4 s_3 s_2$ '. Considering the chunk rule, tags ' $p_3 p_4 p_5$ ' of the POS rule are replaced by the corresponding phrase tag ' c_2 ' as words within the phrase remain consecutive after reordered.

The vocabulary of phrase tags is typically smaller than that of the POS tags. Hence, in order to increase the accuracy of the rules, we decided to use always the POS tag instead of the phrase tag for those phrases composed of a single word. In the previous example, the resulting chunk rule contains the POS tag ' p_6 ' instead of the corresponding chunk tag ' $c3$ '.

Notice from the previous example that an instance of reordering rule is only taken into

and words appear only as leaves, dependency parsing does not postulate non-terminals: words are in bijection with the nodes of the dependency tree. In other words, edges are drawn directly between words. Thus a finite verb has typically an edge directed to its subject, and another to its object.

Figure 4.9 illustrates the constituency (up) and dependency (down) parse trees of the sentence: ‘the boy looks at the man with the telescope’.

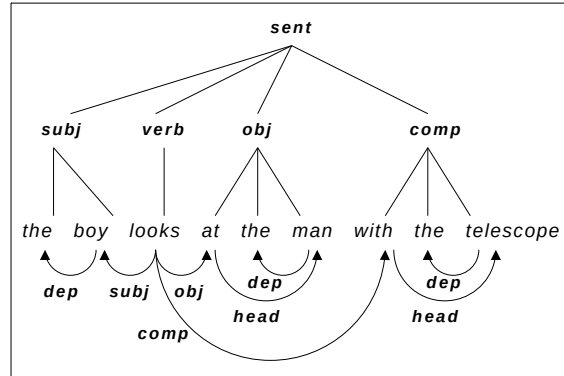


Figure 4.9: Constituency (up) and dependency (down) parsing trees.

As it can be seen, both parsing trees consider the phrase ‘with the telescope’ as a complement of the subject ‘the boy’. However, it can also be considered as a complement of the object ‘at the man’. The example, hence, exhibits the ambiguity of natural languages, one of the most important difficulties that parsing technologies have to deal with.

Next we describe the extension of the reordering rules to account for dependency syntax information.

Figure 4.10 illustrates the process of extracting syntax-based reordering rules. It basically employs the parse trees of the training source sentences (dependency trees) and their word-to-word alignments. *[syntax tree]*, *[zh]*, *[align]* and *[en]* indicate respectively the Chinese sentence dependency tree, the Chinese words, the word-to-word alignment and finally the English corresponding translation sentence. Reordered source words (following the unfold method) are indicated by the *[unfolding]* sequence, where the third source word is moved to the last position.

Once a source reordering is identified, a reordering rule is extracted relating the sequence of words implicated on it. In our example the sequence of words is *[3, 4, 5, 6, 7, 8, 9, 10]*. The procedure to extract a rule from the reordering sequence can be decomposed in two steps:

- The left-hand side of the rule is composed of the dependency structure (subtree of the entire sentence dependency tree) that contains all the words present in the reordering sequence. In Figure 4.11, the structure drawn using bold arcs (top left) shows the left-hand side of the rule.

As it can be seen, the structure relates to the 8 source words implicated in reordering. In some cases, additional source words can be introduced in the rule if they were needed to produce a fully connected subtree. For instance, if the third source word was reordered after the sixth Chinese word, the reordering sequence would initially be *[3, 4, 5, 6]*. However,

the resulting structure would also contain words [7, 8, 9, 10]. The reason is that a fully connected subtree can only be obtained if considering also the additional words.

- Second, nodes of the previous detailed subtree can be pruned out when the source words they relate maintain the same order after applied the reordering rule. See Figure 4.11.

The pruning introduced in the second step is responsible of the generalization power (sparseness reduction) acquired by the syntax-based reordering rules in contrast to the POS-based and chunk-based rules. In our example, Figure 4.10 shows the unpruned rule (bold), and more generalized rules after the successive prunings (labeled *a*) *b*) and *c*)).

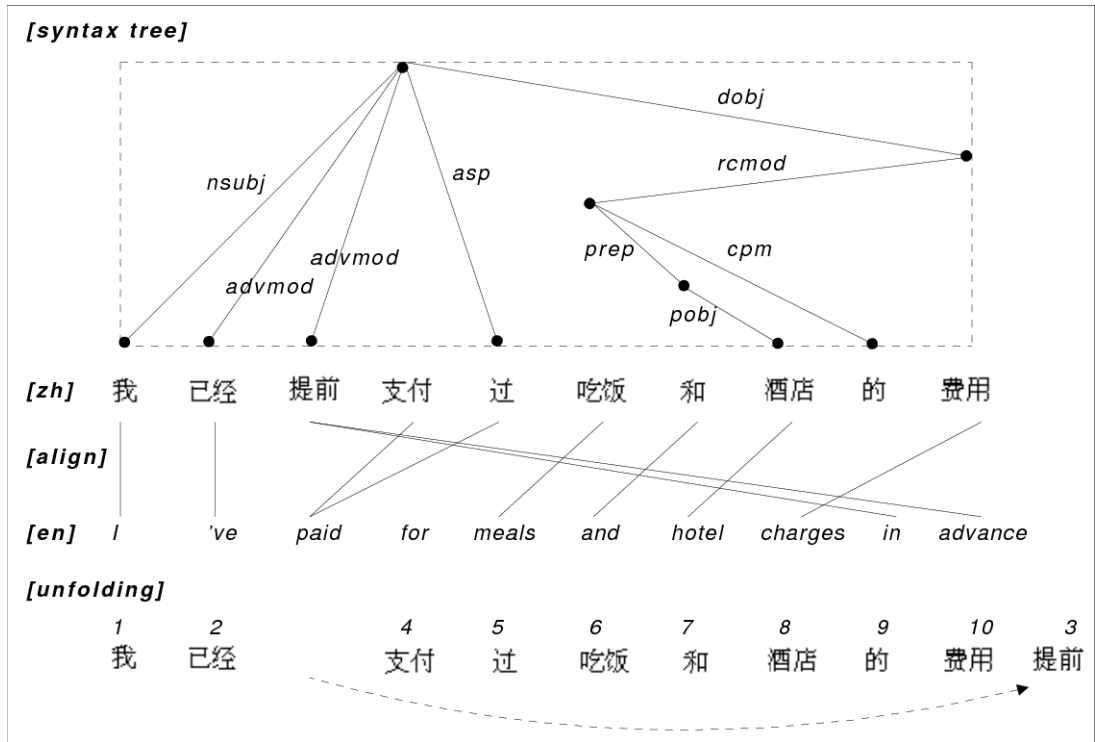


Figure 4.10: Extraction of syntax-based reordering rules. Chinese words are shown in simplified Chinese.

It is worth saying that the generalization power acquired by the pruning method introduces inaccuracy. Some generalized rules are too general and may only be valid in some cases.

The fully-pruned rule (*c*) is internally recorded using the following rule structure:

$$advmod\{1\} \text{ root } asp\{1\} \text{ dobj}\{1\} \rightarrow 1 \ 2 \ 3 \ 0$$

Where nodes (left-hand side of the rule) can designate either words or group of consecutive words, and relationships '*rel*{*x*}' should be read as: current node is a child of node '*x*' under the '*rel*' dependency relationship.

The resulting set of syntax-based rules contains the fully-pruned (generalized with group of words) as well as the unpruned (fully-instantiated) rules. All the rules extracted from the

example are shown in Figure 4.11.

The fully-pruned rules capture a strict superset of the reorderings than are captured by the fully-instantiated rules (at least the same), what makes it redundant to keep all of them. However, the confidence measure of these rules is not the same. As already said, more general rules are also less accurate.

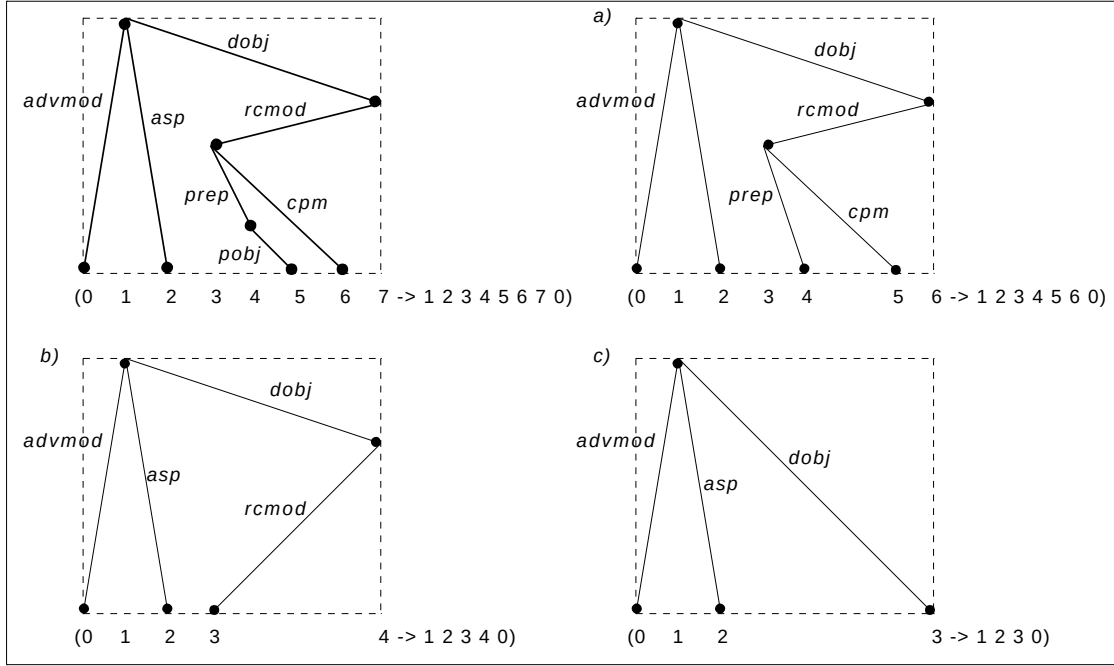


Figure 4.11: Extraction of syntax-based reordering rules. Rule generalization.

4.2.2 Input Graph Extension

In decoding, the input sentence is handled as a word graph. A monotonic word graph contains a single path, composed of arcs covering the input words in the original word order. To allow reordering, the graph is extended with new arcs, which cover the source words in the desired word order.

The motivation of extending the input graph is double: first, the translation quality is aimed at being improved by the ability of reordering following the patterns explained in the previous lines. Second, the reordering decision is more informed since it is taken during decoding using the set of SMT models.

The extension procedure is outlined in the following: starting from the monotonic graph, any sequence of the input POS tags (chunks or dependency subtree) fulfilling a source-side rewrite rule implies the addition of a reordering path. The reordering path encodes the reordering detailed in the target-side of the rule, and is composed of as many arcs as words are present in the pattern.

Figure 4.12 shows an example of reordering graph extension using a POS tag rules. Two

patterns are found in the example, used to extend the monotonic input graph with reordered hypotheses. The example shows (top) the input sentence with POS tags and the monotonic search graph. Then, (middle) the search graph is extended with a reordered hypothesis (dotted arcs) following the reordering pattern ' $NC\ AQ \rightarrow 1\ 0$ ', where the first two words are swapped. Finally, (bottom) it is shown a new extension of the graph following the pattern ' $NC\ AQ\ CC\ AQ \rightarrow 1\ 2\ 3\ 0$ '.

Once the reordering graph is built, it is traversed by the decoder aiming at finding the best translation. Hence, the winner hypothesis is computed using the whole set of system models (fully-informed decision).

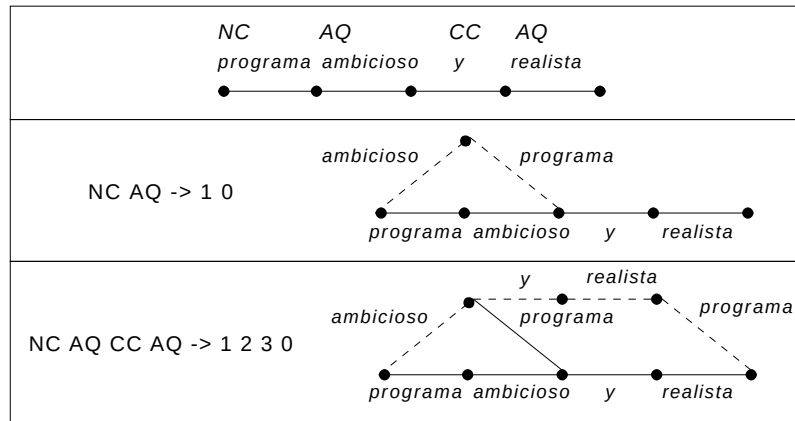


Figure 4.12: Input graph extension.

In the previous example, the input sentence is traversed in decoding ending up in three different sentence word orders:

- *programa ambicioso y realista*
- *ambicioso programa y realista*
- *ambicioso y realista programa*

It is worth to notice that the type of linguistic information used to learn reorderings (either POS tags, chunks or parse trees) does not introduce important differences in the reordering framework employed: patterns are learnt from the same word reorderings introduced in training by the unfold technique. The monotonic input graph is extended using the previous rules, building up a reordered (permutations) graph.

Differences in performance of the reordering framework when employing each linguistic information can only be attributed to the ability to learn (from a training corpus) and produce (over unseen data) valid reorderings of each linguistic source.

4.2.2.1 Recursive Reorderings

Notice that a reordering rule (as detailed in the previous lines) produces always the extension of the monotonic path. This is, the first and last nodes of the new path consist of nodes of the

monotonic path. In other words, the source side of the rules are always matched against the monotonic sequence of words.

The previous particularity of the extension procedure was designed to work with reordering rules built using POS tags, where each tag corresponds exactly with one word. However, when the left-hand side of the rule employs tokens referring to an arbitrary number of words (a chunk or node of a syntax tree), the extension procedure needs to be adapted.

Figure 4.13 justifies the need for the procedure adjustment. POS tags and chunks are shown for the English sentence '*rejected the European Union last referendum*', which is typically translated into Spanish as '*rechazado el último referendun de la Unión Europea*'. The right word order ('*rejected last referendum the Union European*') is obtained after swapping the noun phrases, '*the European Union*' and '*last referendum*', as well as the word sequence '*European Union*'. Hence, following the chunk rule ' $NP\ NP \rightarrow 1\ 0$ ' and the POS rule ' $NN\ NN \rightarrow 1\ 0$ '.

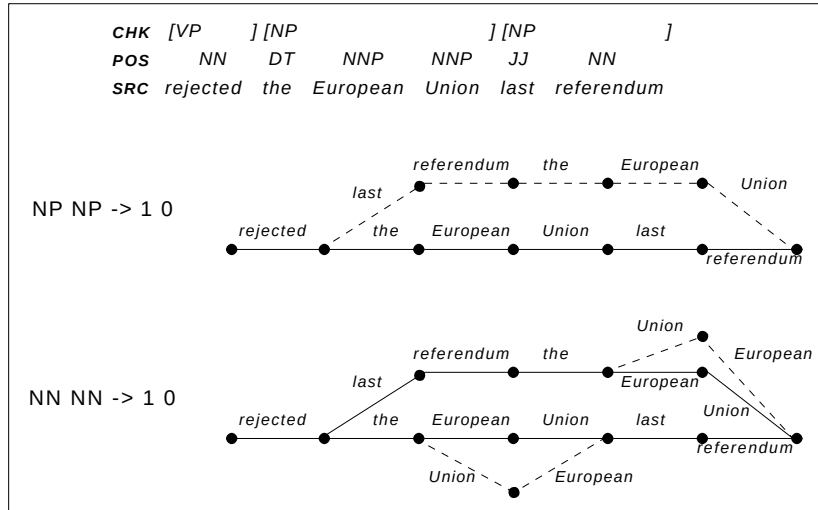


Figure 4.13: Two rules are used to extend the reordering graph of a given input sentence.

However, the extension corresponding to the POS rule (bold arcs) is only obtained when performed on top of the already reordered sequence of words '*last referendum the European Union*'.

Given that chunk (and syntax) rules imply reorderings of sequences of tokens, which may be referred to more than one word (*i.e.* the first chunk ' NP ' is referred to the word sequence '*the European Union*'), further reorderings are sometimes needed to be applied within this tokens (over the internal words of the chunk or subtree).

Accordingly, reordering rules are not only applied on top of the monotonic path but over any sequence of nodes of the reordering graph (recursive reorderings). The input graph extension proceeds extending in order the monotonic path (first the longest reorderings). Introducing first the longest reorderings permits to apply the shortest ones on top of the previous.

4.2.3 Distortion Modeling

We have previously introduced SMT as a double-sided problem: *search* and *modeling*. In previous sections of the current chapter we were concerned with introducing the right reordering hypothesis in the global search. Now, assuming that the right hypothesis can be found in the global search, we have to help the decoder to score it higher than any other. In other words, considering the example of Figure 4.13, we have to use a set of models which score *'rejected last referendum the Union European'* as the most likely path (reordering hypothesis).

In §4.2.1 we introduced a reordering rule probability (see Equation 4.1) used aiming at filtering out noisy patterns. Despite a priori conveying interesting information about the well-likeness of these rules, we discarded the use of this (or any other) information into the reordering graph because of the difficulties of transforming a permutations graph into a weighted permutations graph.

In the introduction section of this chapter we claimed that the proposed reordering approach can take advantage from delaying the reordered decision to the global search, where all the SMT models are available. In accordance, the system already makes use of two models which take care of distortion: the *bilingual N-gram language model* and the *target N-gram language model*. In addition, we introduce two more models to further help the decoder on the reordering task: a *tagged-target N-gram language model*, and a *tagged-source N-gram language model*.

4.2.3.1 Tagged-target N-gram Language Model

This model is destined to be applied over the target sentence (tagged) words. Hence, as the original target language model, computed over raw words, it is also used to score the fluency of target sentences, but aiming at achieving generalization power through using a more generalized language (such as a language of Part-of-Speech tags) instead of the one composed of raw words.

As any N -gram language model, it is described by the following equation:

$$p_{TTM}(s_1^J, t_1^I) \approx \prod_{i=1}^I p(T(t_j) | T(t_{i-N+1}), \dots, T(t_{i-1})) \quad (4.2)$$

where $T(t_j)$ relates to the tag used for the i^{th} target word.

4.2.3.2 Tagged-source N-gram Language Model

This model is applied over the input sentence tagged words. Obviously, this model only makes sense when reordering is applied over the source words in order to monotonize the source and target word order. In such a case, the tagged language model is learnt over the training corpus after reordered the source words.

The new model is employed as a reordering model. It scores a given source-side reordering hypothesis according to the reorderings made in the training sentences (from which the tagged language model is estimated). As for the previous model, source tagged words are used instead of raw words in order to achieve generalization power.

Figure 4.14 illustrates the use of a source and a target POS-tagged N -gram language models. The probability of the sequence '*PRP VRB NN JJ*' is greater than the probability of the sequence '*PRP VRB JJ NN*' for a model estimated over the training set with reordered source words (with English words following the Spanish word order). The opposite occurs considering the tagged-target language model, where the sequence '*VRB JJ NN*' is expected to be highly scored than the sequence '*VRB NN JJ*'⁴.

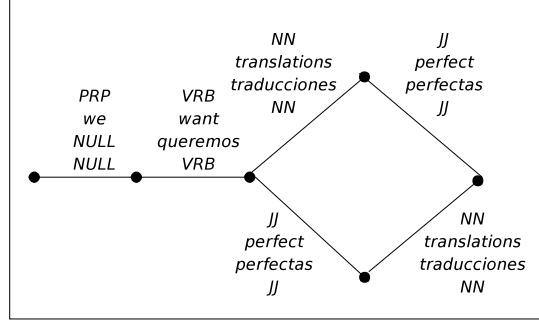


Figure 4.14: Source POS-tagged N -gram language model.

Equivalently, the tagged-source language model is described by the following equation:

$$p_{TSM}(s_1^J, t_1^I) \approx \prod_{j=1}^J p(T(s_j) | T(s_{j-N+1}), \dots, T(s_{j-1})) \quad (4.3)$$

where $T(s_j)$ relates to the tag used for the j^{th} source word.

4.3 Experiments

In this section we detail the experiments carried out to assess the translation accuracy and computational efficiency of the proposed reordering framework. Three different translation tasks are employed, which convey different reordering needs, namely a Spanish-English, an Arabic-English and a Chinese-English tasks. Full details of the corpora employed for the experimentation is shown in A.1.3 (Spanish-English), A.2 (Arabic-English) and A.3 (Chinese-English).

First we give details of several processes common to all tasks.

4.3.1 Common Details

Considering the Spanish-English pair, standard tools were used for tokenizing and filtering. The English side of the training corpus has been POS tagged using the freely available TnT⁵ tagger [Bra00], for the Spanish side we have used the freely available Freeling⁶ tool [Car04].

⁴'VRB', 'JJ' and 'NN' stand respectively for verb adjective and noun.

⁵<http://www.coli.uni-saarland.de/~thorsten/tnt/>

⁶<http://www.lsi.upc.edu/~nlp/freeling/>

Considering Arabic-English, Arabic tokenization was performed following the Arabic Tree-Bank tokenization scheme: 4-way normalized segments into conjunction, particle, word and pronominal clitic. For POS tagging, we use the collapsed tagset for PATB (24 tags). Tokenization and POS tagging are done using the publicly available Morphological Analysis and Disambiguation (MADA) tool [Hab05] together with TOKAN, a general tokenizer for Arabic [Hab06]. For chunking Arabic, we used the AMIRA (ASVMT) toolkit [Dia04]. English preprocessing simply included down-casing, separating punctuation from words and splitting off “’s”. The English side is POS-tagged with the TnT tagger and chunked with OpenNlp⁷, freely available tools.

Considering Chinese-English, Chinese preprocessing included re-segmentation using ICT-CLAS [Zha03]. POS tagging and parsing was performed using the freely available Stanford Parser⁸. English preprocessing includes Part-Of-Speech tagging using the TnT tagger.

After preprocessing the training corpora, word-to-word alignments are performed in both alignment directions using Giza++ [Och03a], and the union set of both alignment directions is computed. Tuple sets for each translation direction are extracted from the union set of alignments. The resulting tuple vocabularies are pruned out considering the N best translations for each tuple source-side ($N = 30$ for the English-to-Spanish direction, $N = 20$ for the Spanish-to-English direction and $N = 30$ for the Arabic-to-English direction) in terms of occurrences.

We used the SRI language modeling toolkit [Sto02]⁹ to compute all N -gram language models (including our special translation model). Kneser-Ney smoothing [Kne95] and interpolation of higher and lower N -grams are always used for estimating the translation N -gram language models.

Once models are computed, optimal log-linear coefficients are estimated for each translation direction and system configuration using an in-house implementation of the widely used downhill SIMPLEX method [Nel65] (detailed in §3.3.3). The BLEU score is used as objective function.

The decoder is always set to perform histogram pruning, keeping the best $b = 50$ hypotheses (during the optimization work, histogram pruning is set to keep the best $b = 10$ hypotheses).

Considering the probability computed for each reordering pattern (see Equation 4.1), all reordering rules (POS-based, chunk-based and syntax-based) which do not achieve a threshold probability $p = 0.01$ are discarded. The value has been empirically set.

4.3.2 Spanish-English Translation Task

In general, Spanish is more flexible with its word order than English is. In both languages, a typical statement consists of a noun followed by a verb followed by an object (if the verb has an object). In English, variations from that norm are used mostly for literary effect. But in Spanish, changes in the word order are very frequently used. It is normally SVO (subject - verb - object), like in *’Juan comió una manzana’* (*’Juan ate an apple’*). However, it is possible to change the word order to emphasize the verb or the object:

The main singularity from the English grammar is that instead of an Adjective-Noun form, Spanish typically follows the Noun-Adjective order. So, in English we would say *’blue car’*, while

⁷<http://opennlp.sourceforge.net/>

⁸<http://nlp.stanford.edu/downloads/lex-parser.shtml>

⁹<http://www.speech.sri.com/projects/srilm/>

in Spanish it would be 'car blue' ('coche azul'). There are exceptions to this rule, particularly when the adjective has a double-meaning.

(VSO)	<i>comió Juan una manzana</i>
(OVS)	<i>una manzana comió Juan</i>
(OSV)	<i>una manzana Juan comió</i>
...	

Figure 4.15: In Spanish the order of the Subject, Verb and Object are interchangeable.

Results

Table 4.1 shows some examples of the Spanish-English reordering patterns extracted using POS tags¹⁰. As it can be seen, patterns are very general rules which may be wrong for some examples. For instance, regarding the sequence of tags 'NC AQ', typically reordered following the pattern 'NC AQ → 1 0', it may be reordered following different rules when appearing within a longer structure (as in 'NC AQ CC AQ → 1 2 3 0').

Table 4.1: Spanish-to-English (top) and English-to-Spanish (bottom) reordering rules.

Reordering rule	Example
NC RG AQ CC AQ → 1 2 3 4 0	<i>ideas muy sencillas y elementales</i>
NC AQ CC AQ → 1 2 3 0	<i>programa ambicioso y realista</i>
NC AQ RG AQ → 2 3 1 0	<i>control fronterizo más estricto</i>
NC AQ AQ → 2 1 0	<i>decisiones políticas delicadas</i>
AQ RG → 1 0	<i>suficiente todavía</i>
NC AQ → 1 0	<i>decisiones políticas</i>
RB JJ CC JJ NN → 4 0 1 2 3	<i>only minimal and cosmetic changes</i>
JJ CC JJ NN → 3 0 1 2	<i>political and symbolic issues</i>
RB JJ JJ NN → 3 2 0 1	<i>most suitable financial perspective</i>
JJ JJ NN → 2 1 0	<i>American occupying forces</i>
NN PO JJ → 2 0 1	<i>Barroso 's problems</i>
JJ NN → 1 0	<i>Italian parliamentarians</i>

A different problem appears when considering the example 'Barroso 's problems'. The sequence is reordered following the pattern 'NN PO JJ → 2 0 1', while the right Spanish word order should be '2 1 0', as it corresponds to the Spanish translation 'problemas de Barroso'. In this case, the reordering rule appears because of bad word alignments in training which forbid learning the right pattern, and reduce the usability of the extracted translation units.

Figure 4.16 illustrates the problem. The link ('s → Barroso) prevents the right unfolding (left). The problem disappears when the right alignments are only used (right). However, the disadvantages of using wrong patterns are reduced because of the fact that translation units are perfectly coupled with the ordering enclosed in patterns. The wrong rule obtains the right translation when employing the tuples extracted also from the wrong alignment.

¹⁰NC; CC; RQ and AQ are Spanish POS tags equivalent to the English POS tags NN; CC; RB and JJ, they stand respectively for noun, conjunction, adverb and adjective.

Tables 4.2 and 4.3 show evaluation results for different experiments considering the Spanish-to-English and English-to-Spanish tasks. The first two rows in Table 4.2 contrast the use of regular (reg) and unfold (unf) translation units. For the system with regular units, monotonic decoding is performed (mon), while a fully reordered search is used for the system with unfold units, constrained to a maximum word distortion limit of three words (*lmax3*). The rest of configurations employ a permutations graph built using POS rules limited to a maximum sequence of *seven* POS tags (graph).

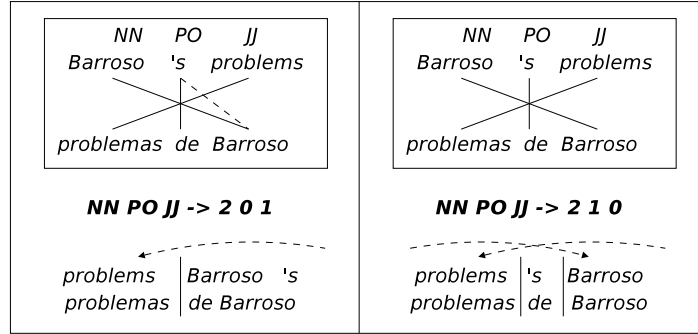


Figure 4.16: Wrong pattern extraction because of erroneous word-to-word alignments.

The second set of experiments (rows three to six) contrast systems considering different N -gram orders for the translation and target language model. Finally, the remaining configurations show the incidence on accuracy of using additional models corresponding to an N -gram language model estimated over the tagged-target words (ttLM) and over the tagged-source words (tsLM). Best scores are shown in bold.

Table 4.2: Evaluation results for experiments with different translation units, N -gram size and additional models. Spanish-to-English translation task.

Units	Search	bLM	tLM	ttLM	tsLM	BLEU	NIST	mWER	PER	METEOR
reg	mon	3	4	—	—	.5556	10.73	34.18	25.17	.6981
unf	lmax3	3	4	—	—	.5231	10.47	37.00	25.32	.6914
unf	graph	3	4	—	—	.5643	10.77	33.58	25.05	.7001
unf	graph	3	5	—	—	.5610	10.74	33.93	25.14	.6994
unf	graph	4	4	—	—	.5616	10.72	33.71	25.29	.6985
unf	graph	4	5	—	—	.5636	10.77	33.58	25.02	.6999
unf	graph	3	4	3	—	.5631	10.76	33.71	25.08	.7021
unf	graph	3	4	4	—	.5658	10.78	33.43	25.07	.7021
unf	graph	3	4	5	—	.5649	10.77	33.52	25.09	.7022
unf	graph	3	4	—	3	.5638	10.75	33.74	25.15	.7002
unf	graph	3	4	—	4	.5674	10.80	33.45	24.99	.7017
unf	graph	3	4	—	5	.5669	10.81	33.37	24.99	.6997
unf	graph	3	4	4	4	.5658	10.79	33.45	25.15	.7013

As it can be seen, both translation tasks show a very similar behavior when contrasting the different configurations.

Regarding the use of regular units under monotonic conditions, it is shown that accuracy results are not far from the best results, which indicates that the considered language pair has limited reordering needs. The fully reordered search (*lmax3*) shows a considerable fall in performance, which is caused by the huge size of the permutations graph (even constrained to a maximum distortion size of three words). As we further detail in the next chapter, our decoder lacks of an estimation cost of the remaining path, what bias the reordering ability into a search for the easiest translated source words.

Table 4.3: *Evaluation results for experiments with different translation units, N -gram size and additional models. English-to-Spanish translation task.*

Units	Search	bLM	tLM	ttLM	tsLM	BLEU	NIST	mWER	PER	METEOR
reg	mon	3	4	—	—	.4793	9.776	41.15	31.52	.6466
unf	<i>lmax3</i>	3	4	—	—	.4449	9.629	42.91	31.55	.6357
unf	graph	3	4	—	—	.4933	9.946	39.79	30.89	.6540
unf	graph	3	5	—	—	.4923	9.938	39.82	30.97	.6535
unf	graph	4	4	—	—	.4934	9.925	39.96	30.84	.6562
unf	graph	4	5	—	—	.4951	9.963	39.78	30.81	.6559
unf	graph	3	4	3	—	.4936	9.898	40.15	31.13	.6556
unf	graph	3	4	4	—	.4954	9.902	40.17	31.03	.6572
unf	graph	3	4	5	—	.4960	9.909	40.27	31.20	.6560
unf	graph	3	4	—	3	.4946	9.944	39.76	30.87	.6553
unf	graph	3	4	—	4	.4957	9.918	39.92	31.00	.6562
<i>unf</i>	<i>graph</i>	<i>3</i>	<i>4</i>	—	<i>5</i>	.4983	9.931	39.81	30.89	.6568
unf	graph	3	4	5	5	.4965	9.896	40.11	31.17	.6573

In both tasks, very similar performance is achieved when considering different N -gram orders for the bilingual and target language models. Considering the additional models, slight improvements are shown (by all measures) when employing the tagged-target language model estimated using 4-grams for the Spanish-to-English task and using 5-grams for the English-to-Spanish task. The tagged-source language model provides a lightly better performance than the tagged-target language model.

Tables 4.4 and 4.5 show evaluation results (using BLEU and mWER scores) for experiments regarding the impact of the maximum size of the POS-based reordering rules for both translation tasks. The best performing systems shown in previous tables (configuration shown in italics) are used for these experiments. No additional optimization work is carried out. Hence, only the impact of the permutation graph is measured in the following experiments. Additionally, it is also shown the number of moves appearing in the 1-best translation option (columns two to seven).

Table 4.6 shows the number of hypothesized reorderings for the test set of each translation task according to their size.

As for the previous experiments, both translation tasks show a similar behavior when contrasting the different configurations. In both cases, the increment in the maximum size of the rules employed to build the permutations graph accounts for accuracy improvements.

As it can be seen, short-distance reorderings are responsible for the most important improvements in accuracy. Rules longer than six words do not introduce further accuracy improvement.

Table 4.4: *Evaluation results for experiments on the impact of the maximum size of the POS-based rules. Spanish-to-English translation task.*

Size	2	3	4	[5,6]	[7,8]	[9,10]	BLEU	mWER
2	1,191	-	-	-	-	-	.5433	35.21
3	1,071	327	-	-	-	-	.5616	33.85
4	1,028	314	142	-	-	-	.5661	33.52
5	1,000	310	120	70	-	-	.5672	33.41
6	994	307	119	89	-	-	.5678	33.39
7	991	306	118	89	13	-	.5674	33.45
8	990	305	118	89	15	-	.5673	33.45
9	990	305	118	89	15	0	.5673	33.45
10	989	305	118	89	15	2	.5671	33.48

This fact can be explained by the (limited) reordering needs of the language pair. A very small number of long-distance moves are captured in the 1-best translation option (4 moves sized from 8 to 10 words for the Spanish-to-English task and 5 moves sized from 8 to 10 words for the English-to-Spanish task).

Table 4.5: *Evaluation results for experiments on the impact of the maximum size of the POS-based rules. English-to-Spanish translation task.*

Size	2	3	4	[5,6]	[7,8]	[9,10]	BLEU	mWER
2	1,647	-	-	-	-	-	.4689	42.20
3	1,424	466	-	-	-	-	.4858	41.00
4	1,355	418	212	-	-	-	.4948	40.08
5	1,330	408	186	76	-	-	.4963	39.92
6	1,315	403	178	119	-	-	.4981	39.78
7	1,295	409	178	108	18	-	.4983	39.81
8	1,313	404	178	113	22	-	.4986	39.80
9	1,313	404	178	113	22	0	.4986	39.80
10	1,313	404	178	113	22	1	.4986	39.80

Table 4.6: *Reorderings hypothesized for the test set according to their size.*

Task	2	3	4	5	6	7	8	9	10
Spanish-to-English	7,599	3,382	2,355	1,431	858	522	277	137	75
English-to-Spanish	8,647	2,811	1,558	1,015	752	510	258	100	37

We have carried out a subjective evaluation of the system reordering ability using 100 translated sentences. We focus on the hypothesized reorderings passed to the decoder, and evaluated as erroneous both, wrong reordering and wrong monotonic decisions. Notice that we do not consider wrong decisions those ending up in wrong translations if the good word order is achieved. For instance, given the input sentence '*programa ambicioso y realista*', the translation '*ambitious and unrealistic program*' is accounted as good even if it is semantically wrong. Results showed that about one out of ten (reordering) decisions were considered wrong.

Despite the inaccuracy of some reordering rules, it seems that for the most of the cases, the set of models employed in the overall search are able to discard the wrong reordering hypotheses.

4.3.3 Arabic-English Translation Task

Arabic is a morpho-syntactically complex language with many differences from English. We describe here three prominent syntactic features of Arabic that are relevant to Arabic-English translation and motivate some of our decisions in this work.

First, Arabic words are morphologically complex containing clitics whose translations are represented separately in English and sometimes in a different order. For instance, possessive pronominal enclitics are attached to the noun they modify in Arabic but their translation precedes the English translation of the noun:

*kitAbu+hu*¹¹ 'book+his' → *his book*'. Other clitics include the definite article *Al+* 'the', the conjunction *w+* 'and' and the preposition *l+* 'of/for', among others.

Separating some of these clitics have been shown to help SMT [Hab06]. In this work we do not investigate which clitics to separate, but instead we use the Penn Arabic Treebank (PATB) [Maa04] tokenization scheme which splits three classes of clitics only. This scheme is compatible with the chunker we use [Dia04].

Secondly, Arabic verb subjects may be: pro-dropped (verb conjugated), pre-verbal (SVO), or post-verbal (VSO). The PATB, as well as traditional Arabic grammar consider the Verb-Subject-Object to be the base order; as such, Arabic VPs always have an embedded subject position. The VSO order is quite challenging in the context of translation to English. For small noun phrases (NP), small phrase pairs in a phrase table and some degree of distortion can easily move the verb to follow the NP. But this becomes much less likely with very long noun phrases that exceed the size of the phrases in a phrase table.

The example in Figure 4.17 illustrates this point. Bolding and italics are used to mark the verb and subordinating conjunction that surround the subject NP (12 words) in Arabic and what they map to in English, respectively. Additionally, since Arabic is also a pro-drop language, we cannot just move the NP following the verb by default since it can be the object of the verb.

[**V AEIn**] [NP-SBJ Almnsq AlEAm lm\$rwE Alskp AlHdyd byn dwl mjls AltEAwn Alxlyjy
 HAmD xAjh] [*SUB An ...*]
 [NP-SBJ The general coordinator of the railroad project among the countries of the Gulf
 Cooperation Council, Hamid Khaja,] [**V announced**] [*SUB that ...*]

Figure 4.17: An example of long distance reordering of Arabic VSO order into English SVO order

Finally, Arabic adjectival modifiers typically follow their nouns (with a small exception of some superlative adjectives). However, English adjectival modifiers can follow or precede their nouns depending on the weight of the adjectival phrase: single word adjectives precede but multi-word adjectives phrases follow (or precede while hyphenated). For example, *rajul Tawiyl* (lit. man tall) translates as 'a tall man', but [*NP rajul [AdjP Tawiyl AlqAmp]*] translates as 'a man tall of stature'.

¹¹All Arabic transliterations in this work are provided in the Buckwalter transliteration scheme [Buc04].

These three syntactic features of Arabic-English translation are not independent of each other. As we reorder the verb and the subject noun phrase, we also have to reorder the insides of the noun phrase adjectival components. This brings new challenges to the previous implementations of N -gram based SMT which had worked with language pairs that are more similar than Arabic and English: although Spanish is like Arabic in terms of its noun-adjective order; Spanish is similar to English in terms of its subject-verb order. Spanish morphology is more complex than English but not as complex as Arabic: Spanish is like Arabic in terms of being pro-drop but has a smaller number of clitics. We do not focus on morphology issues in this work. Table 4.7 illustrates these dimensions of variations. The more variations, the harder the translation. Notice that considering Spanish, the subject-verb as well as the noun-adjective order are not restricted to the ones detailed but they consist of the forms most typically employed.

Table 4.7: Arabic, Spanish and English Linguistic Features

	Morphology	Subj-Verb order	Noun-Adj order
Arabic	hard	VSO, SVO, pro-drop	N-A, A-N
Spanish	medium	SVO, pro-drop	N-A
English	simple	SVO	A-N

As previously stated, the Arabic-English language pair presents important word order disparities. These strong differences make the word alignment a very difficult task, producing typically a huge number of noisy (wrong) alignments. In the case of the N -gram-based approach to SMT, it highly suffers from the apparition of noisy alignments as translation units are extracted out of the single segmentation of each sentence pair.

Noisy alignments typically cause the apparition of large tuples, which imply an important loss of translation information and convey important sparseness problems. In order to reduce the number of wrong alignments, we propose a method to refine the word alignment typically used as starting point of the SMT system. The method employs initially two alignment sets. One with high precision, the other with high recall. We use the **Intersection** and **Union** [Och00a] of both alignment directions (following IBM-1 to IBM-5 models [Bro93]) as high precision and high recall alignment sets respectively.

The method is founded on the fact that linguistic phrases, like raw words, have a translation correspondence and can therefore be aligned. We attempt to make use of chunk information to reduce the number of allowed alignments for a given word. Mainly, we use the idea that words in a source chunk are typically aligned to words in a single target chunk to discard alignments which link words from distant chunks. Considering too strict permitting only one-to-one chunk alignments, we extend the number of allowed alignments by permitting words in a chunk be aligned to words in a target range of words, which is computed as projection of the considered source chunk. The resulting refined set contains all the Intersection alignments and some of the Union.

The algorithm is here outlined. Figure 4.18 shows an example of word alignment refinement. The method can be decomposed in two steps:

First, using the Intersection set of alignments and source-side chunks, each chunk is projected into the target side.

Second, for every alignment of the Union set, the alignment is discarded if it links a source word s_i to a target word t_j that falls out of the projection of the chunk containing the source word. Notice that all the Intersection links are contained in the resulting refined set.

The projection c'_k of the chunk c_k is composed of the sequence of consecutive target words $[t_{left}, t_{right}]$ which can be determined by the next algorithm:

- All target words t_j contained in Intersection links (s_i, t_j) with source word s_i within c_k are considered projection anchors.

In the example, source words of chunk (c_2) are aligned into the target side by means of two Intersection alignments, (s_3, t_3) and (s_4, t_5) , ending up producing two anchors $(t_3$ and $t_5)$.

- For each source chunk c_k , t_{left}/t_{right} is set by extending its leftmost/rightmost anchor, in the left/right direction up to the word before the next anchor (or the first/last word if it does not exist a next anchor).

In the example, c'_1, c'_2, c'_3 and c'_4 are respectively $[t_4, t_4], [t_2, t_6], [t_1, t_2]$ and $[t_6, t_8]$.

In the example, the link (s_1, t_2) is discarded as t_2 falls out of the projection of chunk c_1 ($[t_4, t_4]$).

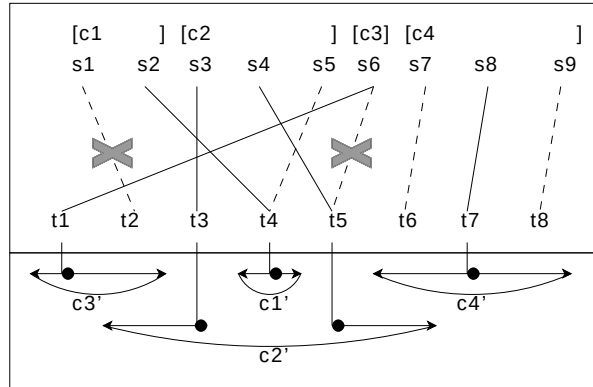


Figure 4.18: Refinement of word alignments using chunks.

A further refinement can be computed considering the chunks of the target side. The same technique applies switching the role of source and target words/chunks in the algorithm. In the second refinement, the links obtained by the first refinement are used as high recall alignment set.

Results

In Table 4.8 we contrast systems built from different word alignments: the Union alignment set of both translation directions (U), the refined alignment set, previously detailed, employing only source-side chunks (rS) and the refined alignment set employing source as well as target-side chunks (rST). Different systems are built considering regular (reg) and unfold (unf) translation

units with accordingly allowing for a monotonic (mon) or a reordered (graph) search. The reordered search is always performed by means of a permutations graph computed with POS-based rules limited to six POS tags. We also assess the order of the bilingual (bLM), tagged-target (ttLM) and tagged-source (tsLM) language models. BLEU and mWER scores are used.

Table 4.8: Evaluation results for experiments on translation units and N -gram size incidence. Arabic-English translation task.

Align	Units	Search	bLM	ttLM	tsLM	MT03		MT04		MT05	
						BLEU	mWER	BLEU	mWER	BLEU	mWER
U	reg	mon	3	—	—	.3785	56.94	.3584	54.23	.3615	55.44
U	unf	graph	3	—	—	.4453	51.94	.4244	50.12	.4366	50.40
rS	unf	graph	3	—	—	.4586	50.67	.4317	49.89	.4447	49.77
rST	unf	graph	3	—	—	.4600	50.64	.4375	49.69	.4484	49.09
rST	unf	graph	4	—	—	.4610	50.20	.4370	49.07	.4521	48.69
rST	unf	graph	5	—	—	.4600	50.91	.4387	49.78	.4499	49.21
rST	unf	graph	4	3	—	.4616	50.74	.4419	49.55	.4502	49.40
rST	unf	graph	4	4	—	.4652	49.94	.4350	49.18	.4533	48.44
<i>rST</i>	<i>unf</i>	<i>graph</i>	<i>4</i>	<i>5</i>	<i>-</i>	.4689	49.36	.4366	48.70	.4561	48.07
rST	unf	graph	4	—	3	.4567	50.97	.4408	49.58	.4472	49.45
rST	unf	graph	4	—	4	.4617	50.51	.4412	49.41	.4519	49.03
rST	unf	graph	4	—	5	.4598	50.56	.4398	49.37	.4518	49.02
rST	unf	graph	4	5	4	.4600	50.75	.4421	49.49	.4506	49.17

A remarkable improvement is obtained by upgrading the monotonic system (first row) with reordering abilities (second row). The improved performance derives from the important differences in word order between Arabic and English. Results from the refined alignment (rS) system clearly outperform the results from the alignment union (U) system. Both measures agree in all test sets. Results further improve when we employ target-side chunks to refine the alignments (rST), although not statistically significantly. BLEU 95% confidence intervals for the best configuration (last row) are ± 0.0162 , ± 0.0210 and ± 0.0135 respectively for *MT03*, *MT04* and *MT05*.

As anticipated, the N -gram system highly suffers under tasks with high reordering needs, where many noisy alignments produce long (sparse) tuples. This can be seen by the increment of translation units when reduced (refined) the number of links, alleviating the sparseness problem by reducing the size of translation units. The number of links of each alignment set consists of $5.5 M$ (U), $4.9 M$ (rS) and $4.6 M$ (rST). Using the previous sets, the total number of extracted units is $1.42 M$ (U), $2.12 M$ (rS) and $2.74 M$ (rST). Accuracy results allow us to say that the refinement technique not only discards alignments but rejects the *wrong* ones.

Extending the translation model to order 4 and introducing the additional 5-gram target-tagged language model (ttLM) seems to further boost the accuracy results. MT04 does not show the same direction, what can be explained by the fact that in contrast to MT03 and MT05, MT04 was built as a mix of topics.

Table 4.9 provides different perspectives of the reordering rules employed to build the permutations graph. Results are obtained through a system featuring a 4-gram translation model with the additional target-tagged 5-gram language model (best system in Table 4.8). Hence, we

now focus on the permutations graph employed as input of the system. Table 4.9 shows for each configuration the total number of sequences where a rule is applied to extend the permutations graph (Total) and the number of moves made in the 1-best translation output according to the size in words of the move (2 to 14) considering only MT03. BLEU scores are also shown for all test sets.

Table 4.9: *Reorderings hypothesized and employed in the 1-best translation output according to their size. BLEU scores are shown for each test set.*

									BLEU	
Size	Total	2	3	4	[5,6]	[7,8]	[9,14]	MT03	MT04	MT05
POS rules										
2	8,142	2,129	-	-	-	-	-	.4364	.4105	.4206
3	2,971	1,652	707	-	-	-	-	.4581	.4276	.4465
4	1,628	1,563	631	230	-	-	-	.4656	.4332	.4532
5	964	1,531	615	210	82	-	-	.4690	.4355	.4549
6	730	1,510	604	200	123	-	-	.4689	.4366	.4561
7	427	1,497	600	191	121	24	-	.4686	.4362	.4562
8	159	1,497	599	191	120	26	-	.4685	.4368	.4565
Chunk rules										
2	9,201	2,036	118	42	20	1	0	.4426	.4125	.4236
3	4,977	1,603	651	71	42	5	2	.4637	.4316	.4507
4	1,855	1,542	593	200	73	7	0	.4680	.4358	.4561
5	1,172	1,514	578	187	118	15	1	.4698	.4381	.4571
6	760	1,495	573	178	130	20	5	.4703	.4373	.4574
7	393	1,488	568	173	129	27	10	.4714	.4372	.4575
8	112	1,488	173	173	129	27	10	.4714	.4373	.4575
7R	-	1,405	546	179	152	54	25	.4725	.4364	.4579

In both cases, configurations consider the kind of rules employed (POS and Chunk), as well as the maximum size of rules allowed to build the graph. 7R indicates that chunk rules are used introducing recursive reorderings. A maximum size of 3 indicates that rules have been used with left-hand side composed of up to 3 POS tags (POS rules) or 3 phrase tags (Chunk rules). Notice that a phrase tag may be referred to multiple words, what explains for instance that 42 moves of size 4 appear using chunk rules of size 2.

As it can be seen, differences in BLEU are very small considering the alternative configurations, what explains that larger reorderings (sized 7 to 14) introduce very small accuracy variations when measured using BLEU. It is shown that POS rules are able to account for the most of the necessary moves (those sized 2 to 6). However, the apparition of the largest moves when considering chunk-based rules (in parallel to accuracy improvements) denotes that long-size reorderings can only be captured by chunk rules (the largest moves taken by the decoder using POS rules consist of 2 sequences of 8 words, no larger moves appear when allowing for larger POS rules).

It is specially relevant the number of long moves when considering recursive chunks (row 7R). This can be understood as that longer chunk rules provide only valid reordering paths if internal word reorderings are also considered. The corresponding BLEU score indicates that the new set of moves improve the resulting accuracy.

Following the example of Figure 4.19 the right reordering path (bold arcs, sized of 11 words) can only be hypothesized by means of the long chunk rule combined with also internal (recursive) reorderings. The figure shows also (bottom) how translation is carried out by composing translation units after reordered the source words. The number shown with each unit indicates the sequence of units (N -gram) shown in training (*i.e.*, the three first units were seen together in the training corpus).

We conducted a human error analysis by comparing the best results from the POS system to those of the best chunk system. We used a sample of 155 sentences from MT03. In this sample, 25 sentences (16%) were actually different between the two analyzed systems. The differences were determined to involve 30 differing re-orderings. In all of these cases, the chunk system made a move, but the POS system only moved (from source word order) in 60% of the cases. We manually judged the relative quality of the move (or lack thereof if the POS did not reorder). We found that 47% of the time, chunk moves were superior to POS choice. In 27% of the time POS was better. In the rest of the time, the two systems were equally good or bad. The main challenge for chunk reordering seems to be the lack of syntactic constraints: in many cases of errors the chunk reordering did not go far enough or went too far, breaking up NPs or passing multiple NPs respectively. Additional syntactic features to constrain the reordering model may be needed.

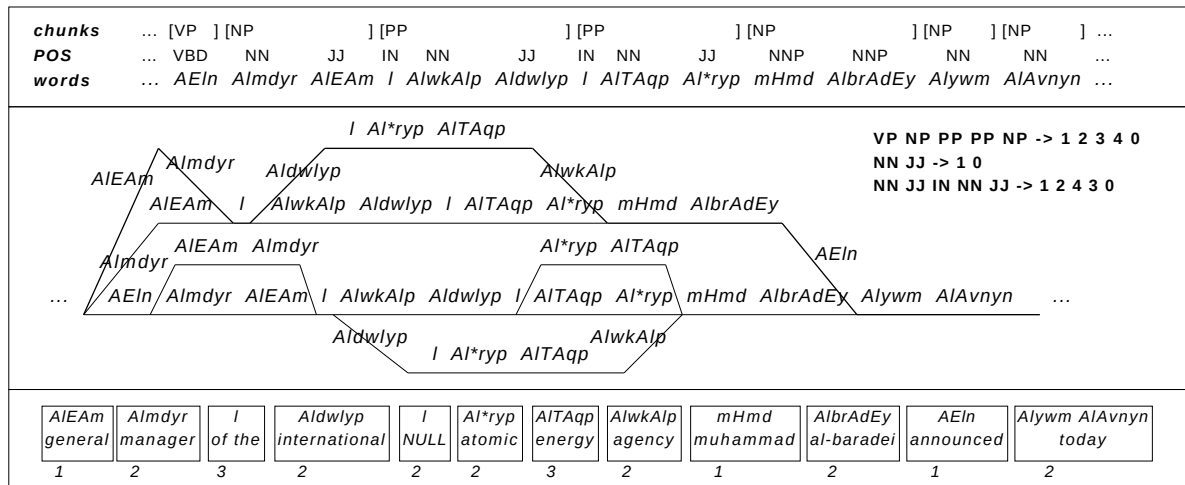


Figure 4.19: Linguistic information, reordering graph and translation composition of an Arabic sentence.

4.3.4 Chinese-English Translation Task

One of the main problems that NLP researchers have to tackle when working with Chinese is the lack of inflectional morphology. Each word has a fixed and single form: verbs do not take prefixes or suffixes showing the tense or the person, number, or gender of the subject. Nouns do not take prefixes or suffixes showing their number or their case. Chinese grammar is mainly concerned with how words are arranged to form meaningful sentences. Hence, word order in Chinese is specially relevant.

The example of Figure 4.20 illustrates this fact. The pair of sentences have identical words

but different meaning because of the word ordering¹²:

The difference in meaning between the two sentences, i.e., definiteness versus indefiniteness of the noun phrases (some person/people versus the person/people) is not expressed by having different words (definite and indefinite articles in English) but by changing the ordering between words.

	<i>sentence₁</i>	<i>sentence₂</i>
zh:	<i>lái rén le</i>	<i>rén lái le</i>
gloss:	come person LE	person come LE
translation:	some person/people have come	the person/people have come

Figure 4.20: Two Chinese sentences with identical words and different meaning ('LE' is an aspect particle indicating completion/change).

Generally, both languages follow the SVO order for the major sentence constituents, i.e., subject preceding verb which in turn precedes the object. However, they differ in many other cases. Next we give a brief overview:

- In Chinese, the modified element always follows the modifier, no matter what kind of modifier it is and how long the modifier is. Figure 4.21 shows two examples with two modifiers, one short, one long, to the same noun. As it can be seen, the noun *shu* in Chinese always occurs at the end of the noun phrase. But in English, the noun *book* occurs at the end of the short noun phrase, but at the beginning of the noun phrase when it contains a long modifier, in this case a relative clause.

	<i>sentence₁</i>	<i>sentence₂</i>
zh:	<i>wo de shu</i>	<i>wo zài shudian mǎi de shu</i>
gloss:	I DE book	I at bookstore buy DE book
translation:	My book	the book I bought at the bookstore

Figure 4.21: Nouns and modifiers in Chinese ('DE' precedes a noun and follows a nominal modifier).

The difference between Chinese and English with respect to the ordering between modifiers and what they modify can be seen also in verbal modifiers. In Chinese all the adverbs and adverbials, which are modifiers for verbs and verb phrases respectively, occur before verbs and verb phrases. But in English, they can occur either before or after verbs or verb phrases. The contrast between English and Chinese can be seen in the possible ways to construct sentences with adverbs and adverbials expressing the same meaning.

- Another difference between Chinese and English has to do with the ordering between noun phrases and prepositions. As the term suggests, prepositions in English occur before noun phrases (hence pre-position), as in *on the table*. In Chinese, however, in addition to prepositions, there are also postpositions, which occur after noun phrases. The prepositions

¹²Chinese examples are provided in *Pinyin*.

and postpositions in Chinese occur sometimes in conjunction with each other, sometimes independent of each other.

Results

In Table 4.10 we contrast systems considering regular (reg) and unfold (unf) translation units with accordingly allowing for a monotonic (mon) or a reordered (graph) search. The reordered search is always performed by means of a permutations graph computed with POS-based rules limited to six POS tags. We also contrast the use of the target (tLM), tagged-target (ttLM) and tagged-source (tsLM) language models. The system features a 3-gram translation language model. BLEU, mWER and METEOR scores are shown.

Table 4.10: *Evaluation results for experiments on translation units and N -gram size incidence. Chinese-English translation task.*

Units	Search	tLM	ttLM	tsLM	dev2			dev3		
					BLEU	mWER	METEOR	BLEU	mWER	METEOR
reg	mon	3	—	—	.4038	45.68	.6180	.4603	40.62	.6615
<i>unf</i>	<i>graph</i>	<i>3</i>	—	—	.4555	38.63	.6294	.5106	34.26	.6711
unf	graph	4	—	—	.4482	39.36	.6269	.5144	34.77	.6711
unf	graph	3	4	—	.4561	39.05	.6306	.5090	34.61	.6725
unf	graph	3	—	4	.4515	39.39	.6340	.5048	35.22	.6760

Mainly, results show a clear improvement when the system introduces distortion (rows 1 and 2), as expected by the language pair. However, slight accuracy differences are shown when contrasting the systems introducing additional models or with models computed for different N -gram order sizes (rows 3 to 5).

Table 4.11 shows the number of POS-based (POS rules), syntax-based (SYN rules) and the union of both (POS+SYN rules) reordering rules hypothesized for the dev2 test set (column Total). Additionally, it is also shown the number of moves (according to their size) introduced in the 1-best translation option (columns 3 to 9) and the corresponding impact on translation accuracy (BLEU scores). Note that no additional optimizations have been carried out. The same model weights for the best system configuration shown in Table 4.10 (second row) is used in all cases. Recursive reorderings are always introduced for SYN and POS+SYN rules.

From the previous table, we can first notice that POS rules introduce less reordering hypotheses than SYN rules.

Considering the shortest rules (*i.e.* rules with two tokens), the SYN approach achieves slightly better results than the corresponding POS approach, in contrast, when longer rules are taken into account, the POS approach slightly outperforms the SYN approach. This situation may be explained by the fact that SYN rules composed of two tokens also account for larger reorderings, which are not considered by POS rules sized 2. When all rules are taken into account, POS rules show higher accuracy results than SYN rules, showing that in general POS rules are more accurate than SYN rules.

When both approaches are used to build a single set of reordering hypotheses (SYN+POS rules), accuracy results are clearly improved for both test sets. We can first affirm that SYN rules

Table 4.11: *Reorderings hypothesized and employed in the 1-best translation output according to their size. BLEU scores are shown for each test set.*

Size	Total	2	3	4	5	6	7	[8, 12]	BLEU	
									dev2	dev3
POS rules										
2	818	157	-	-	-	-	-	-	.4157	.4708
3	622	116	115	-	-	-	-	-	.4234	.4838
4	401	106	87	78	-	-	-	-	.4430	.4982
5	188	99	83	65	36	-	-	-	.4508	.5068
6	55	98	83	63	35	9	-	-	.4555	.5106
7	8	98	83	63	35	9	1	-	.4559	.5105
SYN rules										
2 <i>R</i>	1,518	156	29	17	11	2	2	0	.4169	.4791
3 <i>R</i>	1,206	171	85	23	15	8	4	2	.4285	.4862
4 <i>R</i>	665	151	89	44	22	11	5	4	.4439	.4987
5 <i>R</i>	239	144	84	45	27	15	5	9	.4457	.4973
6 <i>R</i>	59	142	82	44	29	15	4	12	.4509	.5000
7 <i>R</i>	5	142	82	44	29	15	4	12	.4509	.5002
SYN+POS rules										
7 <i>R</i>	-	127	97	72	47	15	6	12	.4714	.5174

have a higher generalization power, as more reorderings are introduced. On the one hand, we can see that rules longer than seven words are only captured by the syntax approach, validating the reduction in sparseness of SYN rules in contrast to POS rules. On the other hand, SYN rules seem to fail on capturing many of the short distance reorderings, given that the combination of both (SYN and POS rules) clearly improves accuracy.

Summing up, short-distance reorderings seem to be better captured by POS rules while long-distance reorderings are only captured by SYN rules.

Notice that the SYN+POS rule set, is a strict superset of the reordering hypotheses introduced by both single approaches (SYN rules and POS rules). The higher accuracy results obtained by the former highlights the remarkable ability shown by the system to employ the set of system models in the overall search to account for the best reordering hypothesis.

4.4 Chapter Summary and Conclusions

This chapter was devoted to the extension of the N -gram-based SMT system with reordering abilities. A reordering framework is detailed which makes use of linguistic information to harmonize the source and target word order. Additionally, using source-reordered translation units provides an interesting way to model reordering by means of the N -gram translation model and also alleviates the data sparseness problem of the translation model caused by using longer units.

We have shown that translation accuracy can be further improved by tightly coupling reordering with the overall search. Hence, reordering decisions are not made solely in a preprocessing

step but during the global search when the whole set of SMT models are available.

Diverse linguistic information sources are studied for the task of learning valid permutations under the reordering framework presented. We have considered the use of part-of-speech, shallow syntax and dependency syntax information. Mainly, using part-of-speech information to account for reordering showed the highest accuracy rates when dealing with short and medium-size reorderings, while it failed on capturing long-distance reorderings. In contrast, shallow and full syntax information provided an interesting method to learn large-distance reorderings at the price of less accurate hypotheses. Interestingly, the combination of part-of-speech and syntactic (either shallow or full) information further outperformed the accuracy results, specially when recursive reorderings were allowed.

In order to model the difference in word order between the source and target languages, the SMT system mainly relies on the N -gram models it includes (bilingual and target language models). Additionally, we have extended the system with two new N -gram models. The first model is applied over the target sentence tagged words. Hence, as the original target language model, computed over raw words, it is also used to score the fluency of target sentences, but aiming at achieving generalization power through using a more generalized language (such as a language of Part-of-Speech tags). The second is applied over the input sentence tagged words. The tagged-source language model is learnt over the training corpus after reordered the source words. Therefore, it scores a given source-side reordering hypothesis according to the reorderings made in the training sentences. As for the previous model, tagged-source words are used instead of raw words in order to achieve generalization power.

Experiments were carried out over three different translation tasks with different reordering needs. Firstly, results obtained for a Spanish-English task showed that short-distance reorderings provided statistically significant improvements using POS-based reordering rules, while no large-distance reorderings appeared necessary. Considering the Arabic-English translation pair, shallow-syntax (chunk) rules offered an interesting tool to overcome the sparseness problem of POS-based rules when dealing with long-distance reorderings. Despite the slight improvement exhibited by automatic measures, a human error analysis revealed the adequacy of chunk-based rules to deal with large reorderings, which were not being captured when using POS-based rules.

Experiments on a Chinese-English translation task showed the adequacy of using dependency syntax to account for the differences in word order of the language pair. Accuracy results outlined the ability of the rules to introduce long-distance reorderings, in special when the long reordering paths include short-distance reorderings too.

Finally, an alignment refinement technique is also detailed that makes use of shallow syntax information to reduce the set of noisy links typically present in translation tasks with important reordering needs. The refinement has been successfully applied on an Arabic-English translation task showing significant improvements measured in terms of translation accuracy.

Chapter 5

Decoding Algorithm for N -gram-based Translation Models

In this chapter we describe a search algorithm, MARIE¹, for statistical machine translation that works over N -gram-based translation models. The chapter is organized as follows:

- In §5.1.2 we review the particularities of the N -gram translation model which motivate singularities in the architecture of the search algorithm when compared to other SMT decoders.
- §5.2 gives details of the algorithm implementation. It follows a beam search strategy based on dynamic programming. Distortion is introduced allowing for arbitrary permutations of the input words, reducing the combinatory explosion of the search space through different constraints and providing an elegant structure to encode reorderings into an input (permutations) graph. The decoder is also enhanced with the ability to produce output graphs, which can be used to further improve MT accuracy in re-scoring and/or optimization work. We report detailed experimental results on search efficiency and accuracy for a large-sized data translation task (Spanish-English).
- In §5.3 we show that apart from the underlying translation model, the decoder also differs from other search algorithms by introducing several feature functions under the well known log-linear framework.
- At the end of the chapter, conclusions are drawn in §5.4.

The decoder has been presented to considerable international translation evaluations as search engine of an N -gram-based SMT system (see Appendix §B).

¹freely available at <http://gps-tsc.upc.edu/veu/soft/soft/marie>. MARIE stands for N -gram-based statistical machine translation decoder.

5.1 Introduction

Research on SMT has been strongly boosted in the last few years, partially thanks to the relatively easy development of systems with enough competence as to achieve rather competitive results. In parallel, tools and techniques have grown in complexity, which makes it difficult to carry out state-of-the-art research without sharing some of this toolkits. Without aiming at being exhaustive, GIZA++², SRILM³ and PHARAOH⁴ are probably the best known examples of freely available toolkits. Accordingly, the piece of code we detail in this chapter constitutes our humble contribution to the set of tools freely available for the SMT community.

5.1.1 Related Work

Statistical machine translation can be seen as a two-fold problem (modeling and search). In accordance, the search algorithm emerges as a key component, core module of any SMT system. Mainly, any technique aiming at dealing with a translation problem needs a decoder extension to be implemented. In general, the competence of a decoder to make use of the maximum of information in the global search is directly connected with the likeliness of successfully improving translations. Accordingly, we describe in detail a decoding algorithm that allows to tackle accurately several translation problems and to couple tightly the overall search with different information sources. We account for the search particularities, which derive from the N -gram-based translation model employed as main feature function.

Experiments of this chapter are performed over the data detailed in Section A.1.3. Translation accuracy results are not given as falling out of the scope of this chapter, which mainly accounts for the search efficiency and accuracy results of algorithm detailed. Further translation accuracy results are given in previous chapters, §3 and §4.

5.1.2 N -gram-based Approach to SMT

Concerning the N -gram-based approach to SMT (detailed in §3 and §4), it can be considered as within (or close to) the phrase-based approach. It employs translation units composed of sequences of source and target words (like standard phrases) and makes use of a beam-based decoder (like phrase-based decoders). However, the modeling of these units (typically called **tuples**) incorporates structural information that makes it necessary the apparition of important singularities in the architecture of the search algorithm.

Like standard phrase-based decoders, MARIE employs translation units composed of sequences of source and target words. In contrast, the translation context is differently taken into account. Whereas phrase-based decoders employ translation units uncontextualized (the translation probability assigned to a phrase unit does not take the surrounding units into account), MARIE takes the translation unit context into account by estimating the translation model as a standard N -gram language model.

Figure 5.1 shows that both approaches (phrase-based and N -gram-based) follow the same generative process, differing in the structure of translation units. In the example, for instance

²<http://www.fjoch.com/GIZA++.html>

³<http://www.speech.sri.com/projects/srilm/>

⁴<http://www.isi.edu/publications/licensed-sw/pharaoh/>

the units ' $s3\#t1$ ' and ' $s1_s2\#t2_t3$ ' of the N -gram-based approach are used considering that both appear sequentially. This fact can be understood as using a longer unit that includes both (longer units are drawn in grey).

Notice that reordering is performed over the source words instead of the source phrases. Thus, translation units with reordered source words are to be considered in the search (the source side word order of these units differs from that of the input sentence). Further details are given in section 5.2.2.

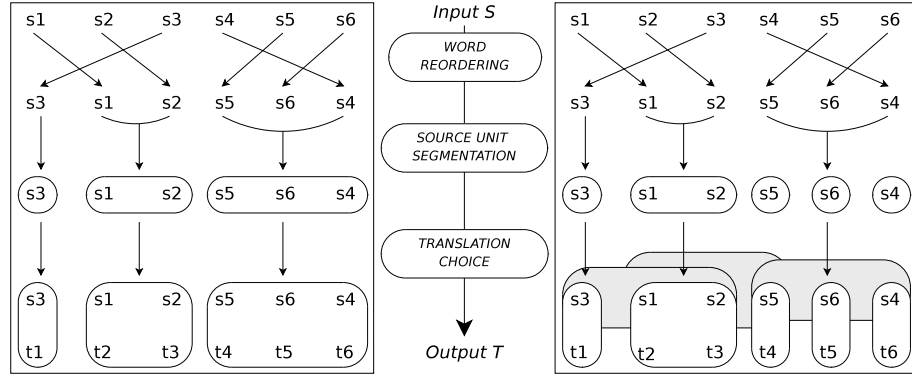


Figure 5.1: Generative process introducing distortion. Phrase-based (left) and N -gram-based (right) approaches.

In the next section we detail the search algorithm. Units (tuples) of the bilingual N -gram translation model are used by the decoder to guide the search. Several additional models are integrated in the log-linear combination expressed in equation 2.6.

5.2 Search Algorithm

As previously stated in §2, SMT is thought as a task where each source sentence s_1^J is transformed into (or generates) a target sentence t_1^I , by means of a stochastic process. Thus, the decoding (search) problem in SMT is expressed by the maximization shown in equations 2.1 and 2.2.

Current SMT systems are founded on the principles of maximum entropy [Ber96]. Under this approach, the corresponding translation of a given source-language sentence s_1^J is defined by the target-language sentence that maximizes a log-linear combination of multiple feature functions $h_i(s, t)$ [Och02], such as described by the following equation:

$$\arg \max_{t_1^I \in \tau} \left\{ \sum_m \lambda_m h_m(s_1^J, t_1^I) \right\}$$

where λ_m represents the coefficient of the m^{th} feature function $h_m(s_1^J, t_1^I)$, which actually corresponds to a log-scaled version of the m^{th} -model probabilities. This equation has been previously introduced in §2.2.

Given that the full search over the whole set of target language sentences is impracticable (τ is an infinite set), the translation sentence is usually built incrementally, composing partial

translations of the source sentence, which are selected out of a limited number of translation candidates (translation units).

The search algorithm implements a **beam search** strategy based on **dynamic programming**. It is enhanced with the ability to perform **reordering**, arbitrary permutations of the input words, and makes use of a **permutation graph** which provides an elegant structure to restrict the number of reorderings in order to reduce the combinatorial explosion of a fully re-ordered search. Like standard SMT decoders it also generates **output graphs** which can be further used in re-scoring and/or optimization work.

Threshold and **histogram** pruning techniques are used to ease the search, as well as **hypothesis recombination**.

Finally, we contrast the structure of the search under N -gram- and phrase-based decoders to highlight the most important singularities of each approach.

5.2.1 Permutation Graph

The decoder introduces reordering (distortion of the input words order) by only allowing the permutations encoded in the input graph (also reordering or permutation graph). Thus, the input graph is only allowed to encode strict permutations of the input words. Any path in the graph must start at the initial node, finish at the ending node and cover the whole input words (without repetitions).

More formally, a word graph is here described as a direct acyclic graph (DAG) $G = (V, E)$ with one root node $n_0 \in V$ and one goal node $n_N \in V$. V and E are respectively the set of nodes and arcs (edges) of the graph G . Arcs are labeled with words and optionally with accumulated scores.

Each node in the graph is marked with a coverage vector, a bit vector of size J representing the source words (where J is the size in words of the input sentence).

A permutation graph has the property that the coverage vector of each node differs from its direct predecessors/successors in exactly one bit. Additionally, nodes are numbered following the linear ordering of a topological sort of the nodes in the graph. It is mathematically proved that DAG's have at least one topological sort. The topological sort guarantees that each hypothesis is visited after all its predecessors [Zen02].

Figure 5.2 shows the reordering graph of an input sentence with 4 words (on top of the figure). Nodes are numbered following the topological sort and labeled with a covering vector of $J = 4$ bits. Each of the full paths (starting at the initial node 0000 and ending at the final node 1111) contains a strict permutation of the input words. Differences between direct predecessors/successors consist in exactly one bit.

The full path '0000, 1000, 1100, 1110, 1111' covers the input words in the original word order. The first arc of the path goes from the initial node 0000 to the node labeled 1000, indicating that the first word is being covered (the difference in their bit vectors is the first bit, which is set to '1' after the transition). Examples of reordering graphs can be seen in figures 5.2, 5.3, 5.4 and 5.10.

Recently, confusion networks have been introduced in SMT [Ber07, Ber05]. In general, confu-

sion networks can be seen as word graphs with the constraint that each full path (from the start node to the end node) goes through all the other nodes. They can be represented as a matrix of words whose columns have different depths (see figure 5.2). The generation of the confusion network from the ASR output word graph may produce in some columns a special word ' ϵ ' which corresponds to the empty word. The use of the ϵ word allows producing source sentences with different number of input words. In contrast, reordering graphs (as presented here) consist of word graphs with the constraint that each full path covers the whole set of words of the input sentence (without repetitions). Hence, each full path differs from each other on the order of the resulting sequence of input words (see figure 5.2).

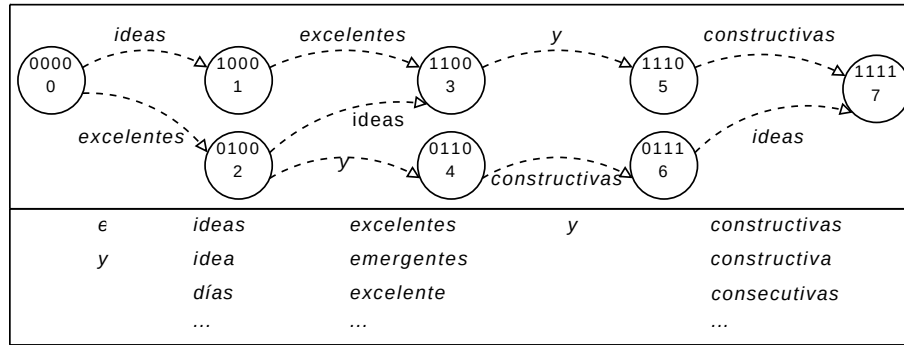


Figure 5.2: Reordering graph (up) and confusion network (down) formed for the 1-best input sentence 'ideas excelentes y constructivas'.

Despite being founded on the same idea, tightly coupling SMT decoding with a preceding process by means of an input graph, it is worth mentioning that reordering graphs and confusion networks neither follow the same objective, nor can be used for the same goals. The former are used to couple reordering and SMT decoding, while the latter aims at coupling speech recognition and machine translation. When decoding the confusion network, one word of each column is picked and used as input word. Thus, input sentences of different length can be hypothesized by using the special ϵ word. However, reordering can not be implemented using the confusion network approach without additional constraints.

The current implementation of the search algorithm can only handle a permutation graph. However, it can be easily extended by removing the permutation constraint to an overall search traversing a more general word graph (without structural constraints). A more general word graph would allow incorporating reorderings as well as different input word options. Hence, making use at the same time of the multiple hypotheses generated by the ASR and reordering preprocessing steps.

5.2.2 Core Algorithm

In the overall search, each node of the input graph is transformed into a stack that contains the set of partial translation hypotheses which cover (translate) the same source words. However, words are not necessarily covered in the same order.

Notice that input reordering graphs differ in several aspects to the overall search graphs. Reordering graphs are used in the search to account for the valid reorderings while search

graphs consist of the data structures of the search hypothesis space.

The core search algorithm is outlined in Algorithm 1. The algorithm takes as input the source sentence (f_1^J), the reordering graph (G), the set of models (h_m) and their weights (λ_h).

Algorithm 1 translate (f_1^J, G, h_m, λ_m)

```

build_units_set(0)
add_hypothesis(NULL, [begin_of_sentence])
for node  $n := 0$  to  $N$  do
  list_tuples := expansion_node[ $n$ ]
  for all  $hyp \in stack[n]$  do
    for all  $tuple \in list\_tuples$  do
      add_hypothesis(& $hyp, tuple$ )
    end for
  end for
end for
trace_back_cheaper(stack[ $N$ ])

```

The search starts by inserting the initial hypothesis (where no source words are yet covered) into the stack labeled with the 0^J bit vector. The hypothesis is used as starting point for the rest of the search algorithm that proceeds expanding all the hypotheses contained on each stack, visiting nodes following the linear ordering of the topological sort (nodes labeled from 0 to N).

Algorithm 2 *build_units_set*(node n)

```

list_tuples :=  $\emptyset$ 
sequences :=  $\emptyset$ 
for all node  $n' \in successors(n)$  do
   $word := arc(n, n')$ 
  sequences' := build_units_set( $n'$ )
  for all sequence  $s \in sequences'$  do
     $s := word.s$ 
    list_tuples := list_tuples  $\cup$  units_with_src_side( $s$ )
  end for
  sequences = sequences  $\cup$  sequences'
end for
expansion_node[ $n$ ] := list_tuples
return(sequences)

```

One of the decoding first steps consists of building the set of translation units to be used in the search. This allows to improve the computational efficiency (in terms of memory size and decoding time) of the search by reducing the look-up time of using a larger translation table. The procedure is outlined in Algorithm 2, implemented following a recursive approach. The procedure is also employed to set the list of tuples that are used to extend the hypotheses of each stack in the search (*list_tuples*).

As it can be seen, when the search introduces reordering, the set of translation options is also extended with those translation units that cover any sequence of input words according to any of the word orders encoded in the input graph (*sequences* in algorithm2).

The extension of the units set is specially relevant when translation units are built from the training set with reordered source words. Typically, a translation table is further constrained by limiting the number of translation options per translation unit source side.

The initial 'empty' hypothesis is specially relevant for N -gram language models (including our special translation model), which also take the beginning and ending of the sentence into account.

The expansion of partial translation hypotheses is performed using new tuples (translation units) translating some uncovered source words. Given the node (or stack) containing the hypothesis being expanded, an expansion is only allowed if the destination node (stack) is a successor, direct or indirect, of the current node in the reordering graph.

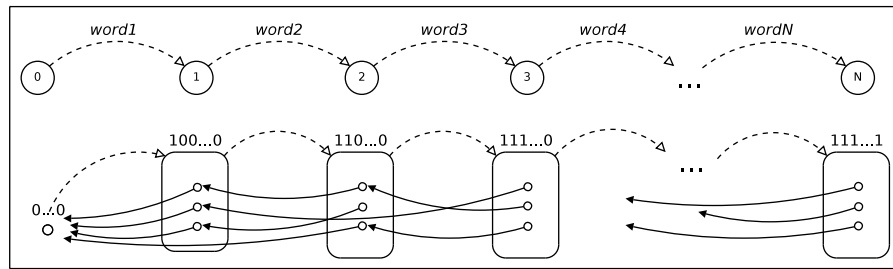


Figure 5.3: Monotonic input graph and its associated search graph for an input sentence with J input words.

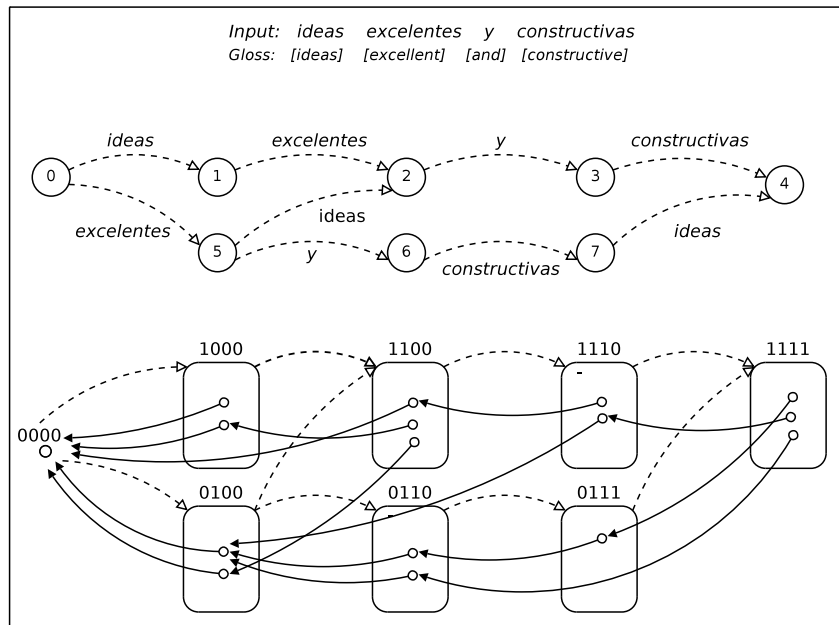


Figure 5.4: Reordered input graph and its associated search graph for the input sentence 'ideas excelentes y constructivas'.

Figures 5.3 and 5.4 illustrate the reordering (up) and search (down) graphs of two different translation examples. The first example is translated under monotonic (figure 5.3) conditions.

Reordering abilities are allowed to translate the second example (figure 5.4).

Dotted arrows are used to draw the arcs of reordering graphs. They point to the successor/s of each node (forward). Regarding search graphs, solid line arrows are used to draw their arcs, which point to each state predecessor (backwards).

Notice that when a partial hypotheses (hyp_i) of the search graph is extended with a translation unit, composing the new partial hypothesis ($hyp_{i'}$), the nodes to which both hypotheses belong (N_{hyp_i} and $N_{hyp_{i'}}$) do not necessarily have to be direct successors/predecessors of each other. This situation is due to the use of translation units with several source words. This is, several input words are being covered at the same time. These units are restricted to follow an available path in the reordering graph (a path starting at node N_{hyp_i} and ending in node $N_{hyp_{i'}}$ must exist). Figure 5.4 illustrates this situation. The first ranked hypothesis in the node labeled '0100' is extended with a hypothesis covering the first and third words at the same time. Hence, stored in the node '1110', which is reachable from node '0100'.

Target words of the translation units are always added sequentially to the target sentence. Thus, building monotonically the target sentence (what makes it easy the use of an N -gram target model score).

Internally, each hypothesis (or state in the overall search) is represented by the set of fields indicated in figure 5.5. The use of additional models in the search introduces additional fields, further discussed in section 5.3.

- A back pointer to the father hypothesis
 - The coverage vector
 - The $N-1$ last tuples used
 - The $N-1$ last target words used
 - The cost so far

Figure 5.5: Fields used to represent a hypothesis.

Every new hypothesis is stored in the stack that contains hypotheses with the same covered source words (described in the covering vector). The hypotheses stored in a stack are sorted according to its accumulated score. It is worth mentioning that in all cases (under monotonic and reordering search conditions) a given hypothesis is allowed to be stored in only one stack.

Under monotonic decoding conditions, the list of tuples (*list_tuples*) to expand a given hypothesis hyp contains those units translating any sequence of consecutive words following the last covered word in hyp (see figure 5.3).

In contrast, under reordering decoding conditions (see figure 5.4), each expansion is allowed to cover any word positions in the source sentence, restricted to be stored in a valid node (or stack) according to the reordering graph.

Every hypothesis is scored with an accumulated score. To compute this score, the cost of the predecessor state is added to the cost derived from the different features used as models. Finally, the translation is output through tracing back the best (lower cost) hypothesis in the last stack ($stack[N]$, where the hypotheses cover the whole input sentence).

Equivalent to the beginning of the sentence, N -gram language models also take the end of

the sentence into account. Therefore, for each hypothesis covering the whole input words (stored in the stack labeled 1^J), the cost derived of the *[end_of_sentence]* token has to be computed and added into its accumulated cost.

5.2.3 Output Graph

Word graphs are successfully used in SMT for several applications. Basically, with the objective of reducing the redundancy of N -best lists, which very often convey serious combinatorial explosion problems.

The goal of using an output graph is to introduce further re-scoring or optimization work. That is, to work with alternative translations to the single 1-best. Therefore, our proposed output graph has some peculiarities that makes it different to the previously sketched input graph.

The structure of arcs remains the same to that of the input graph, but obviously, paths are not forced to consist of permutations of the same tokens (as far as we are interested into multiple translation hypotheses), and there may also exist paths which do not reach the ending node n_N . These latter paths are not useful in re-scoring tasks, but they are shown in order to facilitate the study of the search graph. Furthermore, a very easy and efficient algorithm ($O(n)$, being n the search size) can be used in order to discard them, before re-scoring work. Additionally, given that partial model costs are needed in re-scoring work, our decoder outputs the individual model costs computed for each translation unit.

Multiple translation hypotheses can only be extracted if hypotheses recombinations are carefully saved (recombinations are further detailed in section 5.2.5.2). As outlined in [Koe04], the decoder takes a record of any recombined hypothesis, allowing a rigorous N -best generation. Model costs are referred to the current unit while the global score is accumulated. Notice also that translation units (not words) are now used as tokens.

5.2.4 Contrasting Phrase-based Decoders

In this section we perform a comparison between a system working with tuples (**nbsmt**) and one working with phrases (**pbsmt**). In order to make a fair comparison, both systems were built from the same training data, sharing the decoder as well as the feature functions (obviously with the exception of the translation models). The same accuracy score is achieved by both systems concerning the 1-best translation option. Further details are shown in [Cj07a].

The structure of the search in phrase-based and N -gram-based SMT decoders constitutes an important difference between both approaches.

Regarding the phrase-based approach, the decoder tends to be **overpopulated** of hypotheses consisting exactly of the same translation. This can be explained by the fact that the same translation can be hypothesized following several segmentations of the input sentence. Due to the fact that phrases are collected from multiple segmentations of the training sentence pairs. This problem is somehow minimized by recombining (see section 5.2.5.2) two hypotheses that can not be distinguished by the decoder in further steps. However, the hypotheses can only be recombined (pruned out) once they have been computed, causing the consequent efficiency cost. Additionally, when the decoder generates N -best lists, hypotheses recombination can not be used (see 5.2.5.2), what increases the apparition of multiple equivalent hypotheses.

In order to assess the previous statement we have investigated the list of N -best translations generated by both systems (for the test set). More precisely, The percentage of different translation contained on a given N -best list regarding the size of the list. Figure 5.6 shows the corresponding results for the test set.

It is clearly shown that the N -gram-based approach contains on its N -best list a larger set of (different) translation hypotheses than the phrase-based approach. As it can be seen, the percentage remains close to the 20% for the phrase-based approach. It can be understood as that every 5 translation hypotheses in the N -best list only one different translation is achieved. They consist exactly of the same translation, differently segmented.

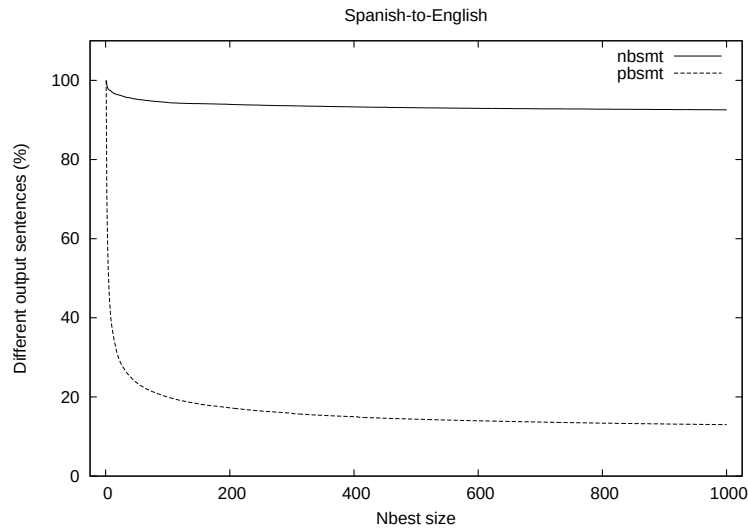


Figure 5.6: *Different translations (%) in the N -best list.*

Results may also be understood as supporting our previous assumption of an **overpopulated** phrase-based search graph because of equivalent translation hypotheses.

We now study the accuracy of the translation options in the N -best lists. In principle, given that the N -best list of the N -gram-based approach contains a larger number of different hypotheses, it will probably contain more accurate translation options than the hypotheses of the phrase-based N -best list.

Figure 5.7 shows the oracle results (measured in WER) regarding the size of the N -best lists. The horizontal lines of the figures indicate the difference in size of the N -best lists (between both approaches) regarding the same oracle score. For instance, to achieve the score $WER = 27$, the phrase-based approach employs an N -best list with 410 more translations.

As it can be seen, the difference in the N -best list size grows exponentially when reducing the oracle score.

On the other hand, the N -gram-based decoder shows a major drawback when compared to standard phrase-based decoders because of a **delayed** probability assignment. That is, an N -gram probability is applied to a translation unit after occupying N valuable positions in different stacks (N translation units), while under the phrase-based approach, the equivalent long phrase is used as a single translation hypothesis [Cj07a].

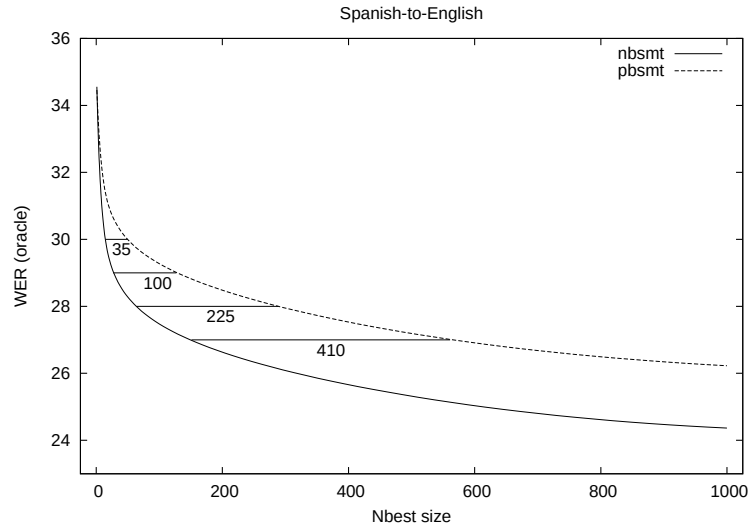


Figure 5.7: Oracle results (WER) regarding the size of the N -best list.

Whenever a long N -gram is matched in the overall search (for instance the N -gram $s1\#t1\ s2\#t2\ s3\#t3$) it typically implies that a long phrase could also be used under a phrase-based approach (the corresponding phrase $s1_s2_s3\#t1_t2_t3$). In such a case, the N -gram of tuples occupies N positions (hypotheses) in different lists of the search ($N = 3$ tuples are used in the example), while only one hypothesis is occupied under the phrase-based approach. Furthermore, despite that the 3-gram probability of the example may be higher than any other N -gram, it does not imply that its initial tuples (the 1-gram ' $s1\#t1$ ' and 2-gram ' $s1\#t1\ s2\#t2$ ') are also highly scored. They could be pruned out in the first lists preventing the 3-gram to appear in further steps of the search (search error).

Summing up, the N -gram-based approach needs a larger search space to hypothesize the same number of translation options than in the phrase-based approach.

Figure 5.8 shows a different point of view to understand the previous interpretation. It consists of an histogram showing the number of translation (final) hypothesis regarding the worst position occupied in the search stacks (beams). It is straightforward shown that the phrase-based translation winner hypotheses tend to occupy higher positions in the beam stacks, all along the search, than the N -gram-based winner hypotheses. Thus, allowing higher values of histogram pruning speeding up the search at no accuracy cost.

The hypotheses situated at the right side of the vertical lines shown in the histograms (labeled $beam = 5, 10$ and 50) represent search errors. That is, winner hypotheses which are lost because of the histogram pruning performed in the search. Search errors are computed from a baseline search performed with a beam size of 100 hypotheses.

Figure 5.9 illustrates the apparition of both effects in the search. On top of the figure it is shown the reordering graph (monotonic search).

As it can be seen, the search under the phrase-based approach is overpopulated because of the method employed to collect translation units from the training corpus.

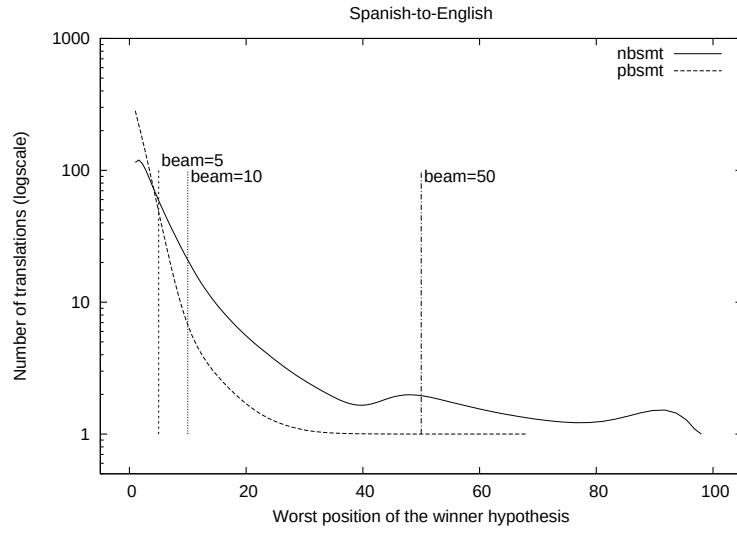


Figure 5.8: *Phrase-based and N -gram-based search errors.*

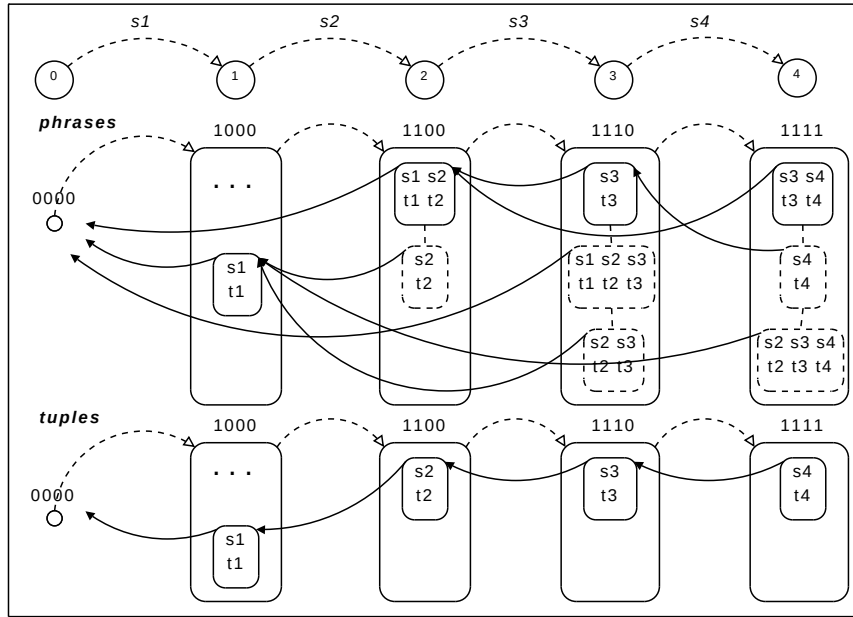


Figure 5.9: *Phrase-based and N -gram-based search graphs.*

On the other hand, regarding the N -gram-based approach, we have that the 3 hypotheses are needed in the search to score the 3-gram ($s1\#t1$ $s2\#t2$ $s3\#t3$), while only one is needed under the phrase-based approach to score the same translation option. Additionally, the three tuples of the 3-gram need to survive to the pruning performed in the three stacks to allow the 3-gram exist in the search.

The overpopulation problem is somehow alleviated by recombining hypotheses (dotted box

hypotheses). The recombination technique is further detailed in section 5.2.5.2.

5.2.5 Speeding Up the Search

The next is an upper bound estimation of the number of hypotheses for an exhaustive search under reordering conditions ⁵:

$$2^J \times (|V_u|^{N_1-1} \times |V_t|^{N_2-1}) \quad (5.1)$$

where J is the size of the input sentence, $|V_u|$ is the vocabulary of translation units, $|V_t|$ is the vocabulary of target words, N_1 the order used in the translation N -gram language model and N_2 the order used in the target N -gram language model.

Despite of the considerably large vocabularies of translation units and target words (in practice the number of different hypotheses within a list is smaller than estimated), the main issue is the exponential complexity of the number of different stacks (2^J), responsible of the NP-completeness of the problem. The estimation can also be read as:

- Different hypothesis stacks (2^J). Also different covering vectors of a fully reordered search. A monotonous search reduces the complexity of this factor to J (instead of 2^J).
- Different hypothesis within a stack ($|V_u|^{N_1-1} \times |V_t|^{N_2-1}$).

Different techniques are used to overcome the complexity problem of this algorithm (which makes the full search unfeasible even for short input sentences). They range from risk-free techniques (hypothesis recombination) to techniques implying a necessary balance between accuracy and efficiency (histogram/threshold pruning, reordering constraints). An additional technique, detailed in section 5.3.6.1, has also been used to reduce the number of access to look-up tables (caching).

5.2.5.1 Reordering Constraints

As introduced in the previous lines, the exponential complexity of the search algorithm is basically produced by the apparition of word reordering. Therefore, the reduction of the whole permutations of a fully reordered search is strictly necessary for even very short input sentences.

The first attempts to reduce the permutations of a reordered search were made by means of different heuristic search constraints. Here, we use '*heuristic*' in opposition to linguistically-based constraints as they are not founded on any linguistic information. Some heuristic reordering constraints are described in [Ber96](IBM), [Wu97](ITG) and deeply analyzed in [Kan05].

Standard reordering constraints can be used with MARIE, encoded into a reordering graph. Figure 5.10 shows a permutation graph built following local constraints ($l = 3$), where the next word to be translated comes from the window of l positions counting from the first uncovered position (in this figure numbers are used instead of source words).

⁵The use of additional models in the global search introduces variations in the previous estimation.

Additionally, the search can also be constrained to a maximum number of reorderings per sentence [Cre05b]. This constraint can only be computed on the fly in the search.

The previous (heuristic) constraints have shown to be useful for some language pairs. They make the search feasible while introducing reordering abilities. However, the use of linguistic information has shown to be a key instrument to account for the structural divergences between language pairs. It is under this latter approach that the use of a word reordering graph becomes of big help as it allows a highly constrained reordered search and a tight coupling between the word ordering and decoding problems.

In phrase-based SMT decoders it is commonly used a further cost estimation strategy [Koe04, Och04b]. This strategy predicts the cost of the remaining path (the cost of the words not yet translated) for each partial hypothesis, and accumulates it into the hypothesis score. The objective of this strategy is to perform a fair comparison between hypotheses covering different words of the input sentence, as phrase-based decoders typically store in the same stack hypotheses covering the same number of input words. Otherwise, the search is biased towards translating first the easiest words (with lower models cost) instead of looking for the right target word order. The decoder bias appears only when stacks are pruned out.

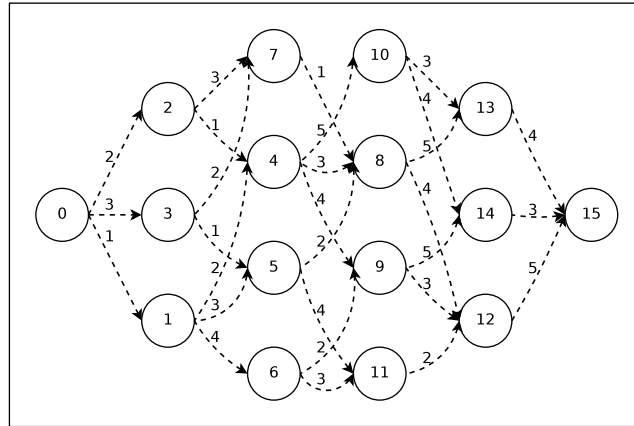


Figure 5.10: *Reordering input graph created using local constraints ($l = 3$).*

The MARIE decoder does not make use of this strategy as only compares (and so prunes) hypotheses covering the same source words. Our strategy can be very expensive in terms of search efficiency under reordering conditions. However, we rely on a very constrained reordered search, thanks to the use of linguistic information to account for the ‘right’ reorderings, which are encoded into a permutation graph that limits the number of reorderings of the search.

Figure 5.11 shows the number of expanded hypotheses (given the number of source words) for different reordering search constraints: **MON**, under monotonic conditions, **RGRAPH**, introducing linguistically-motivated reorderings using an input graph computed from POS-based reordering patterns [Cre06b] and **LOCAL**, allowing reordering using a very limited local (distance-based) constraints, maximum number of reorderings per sentence limited to **three** and a maximum reordering distance of **three** words (**m3j3**).

As it can be seen, the search restricted with reordering patterns achieves a similar level of efficiency than the monotonic search, clearly outperforming the full search with heuristic

constraints. The curves have been smoothed and a log-scale used for the Y-axis.

Regarding accuracy results, in [Cre] is shown that the search constrained with linguistically-motivated reordering patterns clearly outperforms the full search with heuristic constraints. The experiments in [Cre] are carried out over the same training corpus used in this work (slightly differently preprocessed) with different development/test sets.

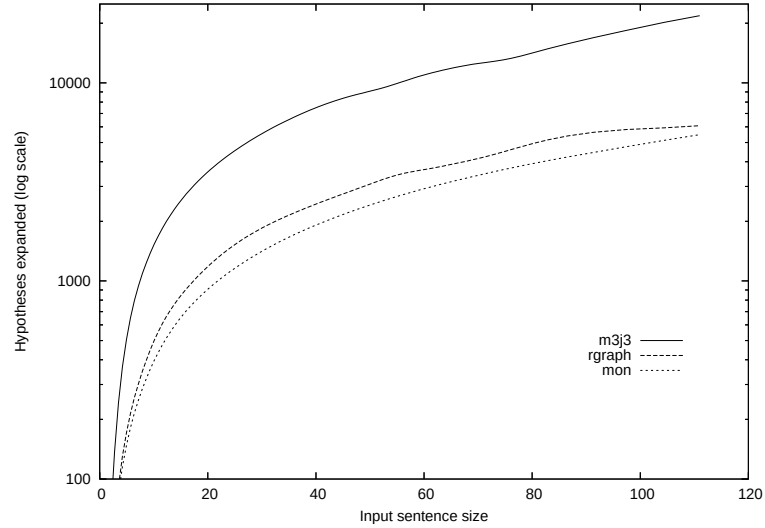


Figure 5.11: *Efficiency results under different reordering conditions.*

5.2.5.2 Hypotheses Recombination

Recombining hypotheses is a risk-free way to reduce the search space when the decoder looks for the single best translation. Whenever two hypotheses cannot be distinguished by the decoder, it automatically discards the one with higher cost (lower probability). Two hypotheses cannot be distinguished by the decoder when they agree in ⁶:

- The last $N_1 - 1$ tuples
- The covering vector (two hypotheses can only be recombined if they belong to the same stack)
- The last $N_2 - 1$ target words (if used the target N -gram language model)

Recombination is risk-free because discarded hypotheses cannot be part of the path containing the best translation. However, when the decoder outputs a word graph (not only the single best translation), it must keep a record of all discarded hypotheses (see below). The reason is that a discarded hypothesis cannot be part of the best translation but of the second one.

⁶The use of additional models in the global search introduces variations to the fields taken into account to recombine hypotheses

As it can be seen in the example of figure 5.9, the phrase-based search graph contains several hypotheses to be recombined, as they can not be distinguished in further steps of the search (drawn using dotted lines and linked to the hypotheses that remains after the recombination). However, when N -best translation options are output, all the hypotheses are kept in the search graph producing multiple equivalent translations (previously discussed in section 5.2.4).

5.2.5.3 Histogram and Threshold Pruning

As already mentioned, under monotonic conditions, hypotheses covering the same source words are stored in the same list. A well known technique to speed up the search consists of discarding the worst ranked hypotheses of each list. Therefore, the size of each list (the beam) in the search can be defined by threshold and histogram pruning.

When expanding a list, only the best hypotheses are expanded: those hypotheses with best scores (**histogram pruning**); with a score within a margin (t) given the best score in the list (**threshold pruning**).

Table 5.1 shows accuracy (search errors) and efficiency (search graph size and decoding time) results for different values of the beam size. Search errors are measured in respect to the translation generated using histogram pruning set to 1000 (without threshold pruning).

Table 5.1: *Histogram pruning (beam size).*

Histogram size	1000	200	100	50	25	10	5
Hypotheses/sent	24,353	5,512	2,520	1,271	646	261	131
Time/sent	38	8.1	4.3	2.0	1.1	0.4	0.2
Search Errors	0%	0.6%	1.9%	4.6%	10%	27%	43%

Table 5.2 shows accuracy (search errors) and efficiency (search graph size) results for different threshold pruning values. Search errors are measured in respect to the translation generated using threshold pruning set to 9 (without histogram pruning).

Table 5.2: *Threshold pruning.*

Threshold value	9	6	5	4	3	2
Hypotheses/sent	24,112	3,415	1,626	729	303	122
Time/sent	33.7	5.6	2.1	1.0	0.4	0.15
Search Errors	0%	0.4%	2.1%	5.8%	19%	45%

As it can be seen, the search accuracy can be kept at reasonable values while speeding up the search by means of the pruning techniques previously detailed.

5.3 Additional Feature Functions

In addition to the tuple N -gram translation model, the N -gram based SMT decoder introduces several feature functions which provide complementary information of the translation process,

namely a target language model, a word bonus model a translation unit bonus model and N additional translation models [Cre07a]. Further details of this features are given in the next sections.

5.3.1 Additional Translation Models

Any additional translation model can be used on the basis of the translation units employed. Standard SMT systems typically use lexical weights to account for the statistical consistency of the pair of word sequences presents on each translation unit.

5.3.2 Target N -gram Language Model

This feature provides information about the target language structure and fluency, by favoring those partial-translation hypotheses which are more likely to constitute correctly structured target sentences over those which are not. The model implements a standard word N -gram model of the target language, which is computed according to the following expression:

$$p_{LM}(s_1^J, t_1^I) \approx \prod_{i=1}^I p(t_i | t_{i-N+1}, \dots, t_{i-1}) \quad (5.2)$$

where t_i refers to the i^{th} target word. The order of the language model can be set up to 9-grams.

5.3.3 Word/Tuple Bonus

The use of any language model probabilities is associated with a length comparison problem. In other words, when two hypotheses compete in the search for the most probable path, the one using less number of elements (being words or translation units) will be favored against the one using more. The accumulated partial score is computed by multiplying a different number of probabilities. This problem results from the fact that the number of target words (or translation units) used for translating a test set is not fixed and equivalent in all paths.

The word bonus and tuple bonus models are used in order to compensate the system preference for short target sentences. They are implemented following the next equations:

$$p_{WB}(s_1^J, t_1^I) = \exp(I) \quad (5.3)$$

where I consists of the number of target words of a translation hypothesis.

$$p_{TB}(s_1^J, t_1^I) = \exp(K) \quad (5.4)$$

where K is the number of translation units of a translation hypothesis.

5.3.4 Reordering Model

We have implemented a 'weak' distance-based (measured in words) reordering model that penalizes the longest reorderings, only allowed when sufficiently supported by the rest of models. It follows the next equation:

$$p_{RM}(s_1^J, t_1^I) = \exp(|j - R(j)|) \quad (5.5)$$

where $R(j)$ is the final position of the source word j (after being reordered).

An additional feature function (distortion model) is introduced in the log-linear combination of equation 5.2:

$$p_{DIST}(u_k) \approx \prod_{i=k_1}^{k_I} p(n_i | n_{i-1}) \quad (5.6)$$

where u_k refers to the k^{th} partial translation unit covering the source positions $[k_1, \dots, k_I]$. $p(n_i | n_{i-1})$ corresponds to the weight of the arc (which links nodes n_i and n_{i-1}) encoded in the reordering graph.

5.3.5 Tagged-target N -gram Language Model

This model is destined to be applied over the target sentence (tagged) words. Hence, as the original target language model (computed over raw words), it is also used to score the fluency of target sentences, but aiming at achieving generalization power through using a more generalized language (such as a language of Part-of-Speech tags) instead of the one composed of raw words. Part-Of-Speech tags have successfully been used in several previous experiments. however, any other tag can be applied.

Several sequences of target tags may apply to any given translation unit (which are passed to the decoder before it starts the search). For instance, regarding a translation unit with the English word '*general*' in its target side, if POS tags were used as target tagged tags, there would exist at least two different tag options: '*NOUN*' and '*ADJ*'.

In the search, multiple hypotheses are generated concerning different target tagged sides (sequences of tags) of a single translation unit. Therefore, on the one side, the overall search is extended towards seeking the sequence of target tags that better fits the sequence of target raw words. On the other side, this extension is hurting the overall efficiency of the decoder as additional hypotheses appear in the search stacks while not additional translation hypotheses are being tested (only differently tagged).

This extended feature may be used together with a limitation of the number of target tagged hypotheses per translation unit. The use of a limited number of these hypotheses implies a balance between accuracy and efficiency. It is estimated as an N -gram language model:

$$p_{TTM}(s_1^J, t_1^I) \approx \prod_{i=1}^I p(T(t_i) | T(t_{i-N+1}), \dots, T(t_{i-1})) \quad (5.7)$$

where $T(t_i)$ relates to the tag used for the i^{th} target word.

5.3.6 Tagged-source N -gram Language Model

The model is applied over the input sentence tagged words. Obviously, this model only makes sense when reordering is applied over the source words in order to monotonize the source and target word order. In such a case, the tagged language model is learnt over the training set with reordered source words.

Hence, the new model is employed as a reordering model. It scores a given source-side reordering hypothesis according to the reorderings made in the training sentences (from which the tagged language model is estimated). As for the previous extension, source tagged words are used instead of raw words in order to achieve generalization power.

Additional hypotheses regarding the same translation unit are not generated in the search as all input sentences are uniquely tagged. It is estimated as an N -gram language model over the source words:

$$p_{TSM}(s_1^J, t_1^I) \approx \prod_{j=1}^J p(T(s_j) | T(s_{j-N+1}), \dots, T(s_{j-1})) \quad (5.8)$$

where $T(s_j)$ relates to the tag used for the j^{th} source word.

In section 5.2.2 was introduced the set of fields that represents a given hypothesis (see figure 5.5). This set is extended with a new element when the target tagged N -gram language model is used in the search. Figure 5.12 shows the extended version of the set of fields.

- A back pointer to the father hypothesis
 - The coverage vector
 - The $N-1$ last tuples used
 - The $N-1$ last target words used
 - The $N-1$ last tagged target words used
 - The cost so far

Figure 5.12: *Extended set of fields used to represent a hypothesis.*

As previously outlined in section 5.2.5, the use of the additional tagged-target N -gram language model and tagged-source N -gram language model introduces variations in the complexity estimation of equation 5.1 and in the fields used to apply the recombination technique. Considering the recombination technique, two hypotheses will now be recombined when they agree in:

- The last $N_1 - 1$ tuples
- The covering vector (two hypotheses can only be recombined if they belong to the same stack)
- The last $N_2 - 1$ target words (if used the target N -gram language model)

- The last $N_3 - 1$ target words (if used the target tagged N -gram language model)
- The last $N_4 - 1$ target words (if used the source tagged N -gram language model)

The complexity estimation of equation 5.1 is extended to the next equation:

$$2^J \times (|V_u|^{N_1-1} \times |V_t|^{N_2-1} \times |V_{Tt}|^{N_3-1} \times |V_{Ts}|^{N_4-1}) \quad (5.9)$$

where V_{Tt} and V_{Ts} are the corresponding vocabularies of tagged target and tagged source words. N_3 and N_4 are the orders used for the corresponding N -gram language models.

Despite the introduction of two new terms in the estimation equation, following the use of two N -gram language models, again, the exponential complexity is derived of the number of different lists (2^J), responsible of the NP-completeness of the problem.

5.3.6.1 Caching

The use of several N -gram language models implies a reduction in efficiency. The singular characteristics of N -gram language models introduce multiple memory access to account for back-off probabilities and lower N -grams fallings.

Many N -gram calls are requested repeatedly, producing multiple calls of an entry. A simple strategy to reduce additional access consists of keeping a record (**cache**) for those N -gram entries already requested. A drawback for the use of a cache consists of the additional memory access derived of the cache maintenance (adding new and checking for existing entries).

Figure 5.13 illustrates this situation. The call for a 4-gram probability (requesting for the probability of the sequence of tokens 'a b c d') may need for up to 8 memory access, while under a phrase-based translation model the final probability would always be reached after the first memory access. The additional access in the N -gram-based approach are used to provide lower N -gram and back-off probabilities in those cases that upper N -gram probabilities do not exist.

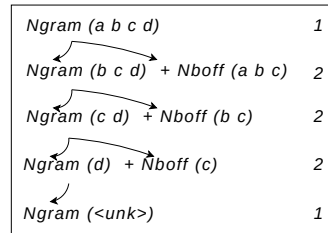


Figure 5.13: Memory access derived of an N -gram call.

Table 5.3 shows translation efficiency results (measured in seconds) given two different beam search sizes. **w/cache** and **w/o cache** indicate whether the decoder employs (or not) the cache technique. Several system configuration have been tested: a baseline monotonic system using a 4-gram translation language model and a 5-gram target language model (**base**), extended with a target POS-tagged 5-gram language model (**+tpos**), further extended by introducing reordering (**+reor**), and finally using a source-side POS-tagged 5-gram language model (**+spos**).

As it can be seen, the cache technique improves the efficiency of the search in terms of decoding time. Time results are further decreased by using more N -gram language models and with a larger search graph (increasing the beam size and introducing distortion).

Table 5.3: *Caching technique results.*

Efficiency	base	+tpos	+reor	+spos
Beam size = 50				
w/o cache	1,820	2,170	2,970	3,260
w/ cache	1,770	2,060	2,780	3,050
Beam size = 100				
w/o cache	2,900	4,350	5,960	6,520
w/ cache	2,725	3,940	5,335	4,880

5.4 Chapter Summary and Conclusions

In this chapter we have presented a search algorithm for statistical machine translation that is specially designed to deal with N -gram-based translation models.

Motivated by the peculiarities of the search architecture and the underlying translation model, remarkable singularities has been shown with respect to standard phrase-based approaches. Mainly, the phrase-based approach allows a higher level of search efficiency while the N -gram-based approach produces higher translations diversity.

Apart from the underlying translation model, the decoder contrasts to other search algorithms by introducing several feature functions under the well known log-linear framework and by a tight coupling with source-side reorderings. The combinatory explosion of the search space when introducing reordering can be easily tackled through encoding reorderings into an input (permutations) graph.

The search structure permits a highly fair comparison of hypotheses before pruning when compared to standard phrase-based decoders, as hypotheses stored in a stack translate exactly the same input words. This makes unnecessary the use of a further cost estimation strategy typically used in phrase-based decoders. Our strategy can be computationally very expensive. However, the use of linguistic information can highly constraint the set of reorderings, achieving search efficiency results close to monotonic conditions.

The decoder is also enhanced with the ability to produce output graphs, which can be used to further improve MT accuracy in re-scoring and/or optimization work. Finally, we have shown a caching technique that alleviates the cost of the additional table look-ups produced by the N -gram language models.

The current implementation of the search algorithm can only handle a permutation graph. However, it can be easily extended to explore a more general word graph, without being restricted to be a permutation graph. A more general word graph would allow incorporating reorderings as well as different input word options. Thus, integrating in a single word graph the multiple hypotheses generated by an ASR as well as the multiple reordering paths.

Chapter 6

Conclusions and Future Work

This PH.D. dissertation has considered the fully description of the N -gram-based approach to SMT. In special, we participated in the initial monotonic version definition and upgraded the system with reordering abilities. The following scientific contributions have been achieved:

- We have participated in the definition and implementation of many of the features and strategies employed in the N -gram-based system. Among others we can mention the translation unit definition (extraction and refinement). Thanks to the many changes introduced, the SMT system has grown achieving comparable results to other outstanding systems. Full details of this contribution are given in Chapter 3.
- We have described an elegant and efficient approach to introduce reordering into the SMT system. The reordering search problem has been tackled through a set of linguistically motivated rewrite rules, which are used to extend a monotonic search graph with reordering hypotheses. The extended graph is traversed during the global search, when a fully-informed decision can be taken. Different linguistic information sources have been considered and studied. They are employed to learn valid permutations under the reordering framework introduced. Despite that the reordering framework has been applied all along this work to an N -gram-based SMT system, it can also be considered for standard phrase-based systems. Full details of this contribution are given in Chapter 4.
- Additionally, a refinement technique of word alignments is presented which employs shallow syntax information to reduce the set of noisy alignments present in an Arabic-English task. Full details are given in Chapter 4.
- We have described a search algorithm for statistical machine translation that is specially designed to work over N -gram-based translation models, where the bilingual translation history is differently taken into account than in standard phrase-based decoders. Considering reordering, it allows to introduce distortion by means of an input graph where arbitrary permutations of the input words are detailed. Therefore providing a tight coupling between reordering and decoding tasks. The decoder is also enhanced with the ability to produce output graphs, which can be used to further improve MT accuracy in re-scoring and/or optimization work. The decoder also differs from other search algorithms by introducing several feature functions under the well known log-linear framework. Full details of this contribution are given in Chapter 5.

6.1 Future Work

Several lines for future research are envisaged, which can extend the work presented in this Ph.D. dissertation. Among others, we can mention:

- Use of an unrestricted input graph. In this thesis work we have shown that translation accuracy can be improved by tightly coupling reordering and decoding tasks. following this direction the permutations graph can be enhanced with the ability to decode unrestricted input graphs. This tiny extension would give us a powerful tool to tackle several additional problems: it would allow to decode the N-best recognition hypotheses of an ASR system, input sentences could be built following different tokenization hypotheses (specially relevant for languages such as Chinese), out-of-vocabulary words could be replaced by several word alternatives. More generally, we can provide the overall search with alternative word/phrase/idiom hypotheses which are equivalently translated into the target language but with a higher level of representativity in the translation model (with probabilities more robustly computed). The idea relies on the extremely huge amounts of monolingual data available, while small size (and expensive) parallel corpora is available. Monolingual data can be used to analyze the input words/structure and convert them into semantically equivalent hypotheses which be easily translated making use of the available translation model.
- Further boosting the use of linguistic information in the translation process. Current translation units are merely based on the brute force of computers, which produce translations as a composition of raw translation pieces (commonly called phrases) previously seen in a training corpus. Even for extremely large corpus, results are only acceptable when testing systems on closely related data, leaving a lot to be desired when moving away from it. Intelligently replacing raw words by linguistic classes would alleviate some of the difficulties of current SMT systems, such as the sparseness problem in translation models, the modeling of long-distance discontinuities or the difficulties to deal with erroneous or out-of-domain data.
- Tight coupling the three technologies implied in a Speech-to-Speech translation system. SMT is typically carried out for a single-best ASR recognition hypothesis. However, it is already shown that SMT accuracy can be improved by allowing to translate N-best recognition hypotheses instead of the single-best. Considering the TTS system, under a Speech-to-Speech context, the quality of the output speech can also be improved by carrying some of the features contained on the input speech, which need to be synchronized with the translated text.

Appendix A

Corpora Description

For all corpora used in this work, the training data is preprocessed by using standard tools for tokenizing and filtering. In the filtering stage sentence pairs are removed from the training data in order to allow for a better performance of the alignment tool. Sentence pairs are removed according the following two criteria:

- Fertility filtering: removes sentence pairs with a word ratio larger than a predefined threshold value.
- Length filtering: removes sentence pairs with at least one sentence of more than 100 words in length. This helps to maintain alignment computational times bounded.

Next we detail the corpora used all along this thesis work. Tables present the basic statistics for the training, development and test data sets for each considered language. More specifically, the statistics show the number of sentences, the number of words, the vocabulary size (or number of distinct words) and the number of available translation references (M and k stand for millions and thousands).

A.1 EPPS Spanish-English

The EPPS data set is composed of the official plenary session transcriptions of the European Parliament, which are currently available in eleven different languages [Koe05b].

All the experiments in this work are carried out over the *Final Text Edition* version of the corpus (FTE). It mainly consists of text transcriptions of the Parliament speeches after edited and rewritten in some cases in order to include punctuation, truecase and avoid different spontaneous speech phenomena.

Evaluation experiments are presented considering different versions of the corpora released for the different TC-Star evaluations.

A.1.1 EPPS Spanish-English ver1

It consists of the Spanish and English versions of the EPPS data that have been prepared by RWTH Aachen University in the context of the European Project TC-STAR¹. Table A.1 shows the basic statistics.

Table A.1: *EPPS ver1. Basic statistics for the training, development and test data sets*

Set	Language	Sentences	Words	Vocabulary	References
Train	Spanish	1.22 M	34.8 M	169 k	-
	English	1.22 M	33.4 M	105 k	-
Dev.	Spanish	1,008	25.7 k	3.9 k	3
	English	1,008	26.0 k	3.2 k	3
Test	Spanish	840	22.7 k	4.0 k	2
	English	1,094	26.8 k	3.9 k	2

A.1.2 EPPS Spanish-English ver2

This version introduces additional (in-domain) training data to the previously detailed corpus and differs on the tokenization employed on both, English and Spanish words. As it can be seen, apart from the entire corpus (full) two training subsets are considered (medium and small) which consist of the first (100k and 10k) sentence pairs of the entire training corpus. Table A.2 shows the basic statistics.

Table A.2: *EPPS ver2. Basic statistics for the training, development and test data sets.*

Set	Language	Sentences	Words	Vocabulary	References
Train (full)	Spanish	1.28 M	36.6 M	153 k	-
	English	1.28 M	34.9 M	106 k	-
Train (medium)	Spanish	100 k	2.9 M	49.0 k	-
	English	100 k	2.8 M	34.8 k	-
Train (small)	Spanish	10 k	295 k	17.2 k	-
	English	10 k	286 k	12.7 k	-
Dev.	Spanish	430	15,3 k	3.2 k	2
	English	735	18,7 k	3.1 k	2
Test	Spanish	840	22,7 k	4.0 k	2
	English	1,094	26,8 k	3.9 k	2

¹TC-STAR (Technology and Corpora for Speech to Speech Translation)

A.1.3 EPPS Spanish-English ver3

This last version of the EPPS training corpus slightly differs from the previous by introducing additional material and by the tokenization employed on source and target words. Table A.3 shows the basic statistics.

Table A.3: *EPPS ver3. Basic statistics for the training, development and test data sets.*

Set	Language	Sentences	Words	Vocabulary	References
Train	Spanish	1.27 M	36.1 M	152 k	-
	English	1.27 M	34.5 M	105 k	-
Dev.	Spanish	1,008	25.7 k	3,9 k	2
	English	1,008	26.0 k	3,2 k	2
Test	Spanish	840	22.7 k	4,0 k	2
	English	840	26.8 k	3,9 k	2

A.2 NIST Arabic-English

All of the training data used is available from the Linguistic Data Consortium (LDC²). The parallel text includes Arabic News (LDC2004T17), eTIRR (LDC2004E72), English translation of Arabic Treebank (LDC2005E46), and Ummah (LDC2004T18). For tuning and testing we used the standard four-reference NIST MTEval datasets for the years 2002, 2003, 2004 and 2005. Table A.4 presents the basic statistics of training, tuning and test data sets for each considered language.

Table A.4: *NIST Arabic-English corpus. Basic statistics for the training (train), development (MT02) and test data sets (MT03, MT04, MT05).*

Set	Language	Sentences	Words	Vocabulary	References
Train	Arabic	130.5 k	4.1 M	72.8 k	-
	English	130.5 k	4.4 M	65.9 k	-
MT02	Arabic	1,043	29.1 k	5,9 k	4
MT03	Arabic	663	18.3 k	4,3 k	4
MT04	Arabic	1,353	42.1 k	8,4 k	4
MT05	Arabic	1,056	32.1 k	6,3 k	4

²<http://www ldc upenn edu/>

A.3 BTEC Chinese-English

The Chinese-English data employed here consists of sentences randomly selected from the BTEC³ corpus [Tak02]. Tuning and test sets correspond to the official CSTAR03, IWSLT04 and IWSLT05 evaluation data sets⁴. Table A.5 presents the basic statistics of training, tuning and test data sets for each considered language.

Table A.5: *BTEC Chinese-English corpus. Basic statistics for the training (train), development (dev1) and test data sets (dev2, dev3).*

Set	Language	Sentences	Words	Vocabulary	References
Train	Chinese	39.9 k	342.1 k	11.2 k	-
	English	39.9 k	377.4 k	11.0 k	-
dev1	Chinese	506	3.3 k	880	16
dev2	Chinese	500	3.4 k	920	16
dev3	Chinese	506	3.7 k	930	16

³Basic Travel Expression Corpus

⁴<http://iwslt07.itc.it/>

Appendix B

Participation in MT Evaluations

International evaluation campaigns have supposed an important factor of the impressive growth of SMT in the last few years. Organized by different institutions, consortiums, conferences or workshops, these campaigns are the perfect instrument to assess the translation improvements of different SMT systems. Furthermore, systems are fairly compared and knowledge is shared among researchers from several research institutions.

With a large experience in automatic speech recognition benchmark tests, the National Institute of Standards and Technology (NIST), belonging to the Government of the United States, organizes yearly machine translation tests since the early 2000s. Aiming at a breakthrough in translation quality, these tests are usually unlimited in terms of data for training. The target language is English, and sources include Arabic and Chinese¹.

Since October 2004, the C-STAR² consortium organizes the *International Workshop on Spoken Language Translation* (IWSLT) on a yearly basis. This workshop includes an evaluation campaign oriented towards speech translation and small data availability. Therefore, training material tends to be limited. Language pairs include Chinese, Japanese, Korean, Arabic, Italian and English (usually English being the target language). Reports of the 2005, 2006 and 2007 editions are published in [Aki04] and [Eck05], respectively³.

In 2005, a Workshop on *Building and Using Parallel Texts: data-driven MT and beyond*, organized at the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), also included a machine translation shared task reported in [Koe05c]. In this case, translation between European languages (Spanish, Finnish, French, German and English) was the main task. Training included the European Parliament proceedings corpus [Koe05b]. In 2006, a new edition of this evaluation campaign was conducted in the HLT/NAACL'06 *Workshop on Statistical Machine Translation*, as reported in [Koe06]. Finally, the last edition of this evaluation was organized in the ACL'07 *Second Workshop on Statistical Machine Translation*, reported in [CB07]. In this last evaluation shared task four language pairs in both directions were included (English-German, English-French, English-Spanish and English-Czech).

Additionally, the European project TC-STAR (Technology and Corpora for Speech to Speech Translation) organized a first internal evaluation in 2005 (for members of the project, including

¹<http://www.nist.gov/speech>

²Consortium for Speech Translation Advanced Research, <http://www.c-star.org>

³<http://www.is.cs.cmu.edu/iwslt2005> - <http://www.slc.atr.jp/IWSLT2006> - <http://iwslt07.itc.it>

UPC) and an open evaluation in 2006 and 2007⁴.

In the next sections are presented the results achieved by the UPC N -gram-based SMT system in several international evaluation campaigns.

B.1 TC-Star 3rd Evaluation

The TC-Star EU-funded project organized its last evaluation on February 2007. Language pairs included English-Spanish, in which UPC took part, and Chinese-English. Roughly speaking parallel training data consisted of the European Parliament corpus.

To study the effect of recognition errors and spontaneous speech phenomena, particularly for the EuParl task, three types of input to the translation system were studied and compared:

- **ASR**: the output of automatic speech recognizers, without using punctuation marks
- **verbatim**: the verbatim (i.e. correct) transcription of the spoken sentences including the phenomena of spoken language like false starts, ungrammatical sentences etc. (again without punctuation marks)
- **text**: the so-called final text editions, which are the official transcriptions of the European Parliament and which do not include the effects of spoken language any more (here, punctuation marks were included)

In addition to these tasks, a complementary Spanish to English task was included in this evaluation for portability assessment. This data consisted of transcriptions from Spanish Parliament, for which no parallel training was provided. Further details on the evaluation can be read in the evaluation website mentioned in the lines above.

The N -gram-based SMT system presented to the evaluation was built from *unfold* translation units, and making use of POS-tag rules to account for reorderings. A set of 6 additional models were used: a target language model, a word bonus, a target tagged language model, a source (reordered) language model and two lexicon models computed on the basis of word-to-word translation probabilities.

A Spanish morphology reduction was implemented in the preprocessing step, aiming at reducing the data sparseness problem due to the complex Spanish morphology. In particular, Spanish pronouns attached to the verb were separated, i.e. '*calculamos*' is transformed into '*calcula +mos*'. And contractions like '*del*' were also separated into '*de el*'. GIZA++ alignments were performed after the preprocessing step.

Tables B.1 and B.2 detail respectively the Spanish to English and English to Spanish results (in terms of the automatic measures BLEU and NIST). Table B.1 contains accuracy results for the two corpus domains used in the evaluation (*Euparl* and *cortes*). Results consider three different tasks for each translation direction (*FTE*, *Verbatim* and *ASR*).

Notice that in the official results of the evaluation, multiple submissions for each participant and task are considered as well as a system combination '*ROVER*' that we have not introduced

⁴<http://www.elda.org/tcstar-workshop/2007eval.htm>

in this summary (we picked the best submission results of each participant when multiple were available).

Table B.1: *TC-Star’07 Spanish-English automatic (BLEU/NIST) comparative results for the three tasks (FTE, Verbatim and ASR) and corpus domains (Euparl and Cortes). Site Rank is shown in parentheses.*

Site	Task	<i>FTE</i>		<i>Verbatim</i>		<i>ASR</i>	
		BLEU	NIST	BLEU	NIST	BLEU	NIST
IBM	Euparl	0.5406 (1)	10.77	0.5508 (1)	10.89	0.4265 (1)	9.630
ITC-irst		0.5240 (4)	10.56	0.5208 (3)	10.55	0.3793	9.210
RWTH		0.5310 (2)	10.65	0.5506 (2)	10.94	0.3944 (2)	9.380
UED		0.5187	10.48	-	-	-	-
UKA		0.4705	9.980	0.4600	9.850	0.3302	8.530
UPC		0.5230 (5)	10.60	0.5200 (4)	10.45	0.3833 (3)	9.150
DFKI		0.4304	9.470	0.4220	9.330	-	-
UW		0.5261 (3)	10.53	0.4786	9.850	-	-
SYSTRAN		0.4572	9.720	0.4528	9.680	0.3379	8.850
LIMSI		-	-	0.4599	9.760	0.3360	8.710
IBM	Cortes	0.4208 (1)	9.260	0.5014 (1)	10.20	0.3606 (1)	8.710
ITC-irst		0.3966	8.960	0.4570	9.680	0.3053	0.080
RWTH		0.4092 (2)	9.130	0.4988 (2)	10.25	0.3270 (2)	8.340
UED		0.3904	8.850	-	-	-	-
UKA		0.3517	8.450	0.4045	9.110	0.2712	7.630
UPC		0.4037 (3)	9.060	0.4728 (3)	9.910	0.3119 (3)	8.140
DFKI		0.3110	7.910	0.3282	8.180	-	-
UW		0.3830	8.760	0.4213	9.110	-	-
SYST		0.3502	8.320	0.4240	9.260	0.2848	7.860

Considering the Spanish to English results, the UPC system achieves very competitive results when compared to other participants. It is remarkable that our system is better ranked when the translation domain of the test data gets far from the domain employed to train the system. The system is better ranked not only when moving from *Euparl* to *Cortes*, but also from *FTE* to *Verbatim* and *ASR*. So far a single system is used for all tasks, which was built from data in the form of *FTE*, *Verbatim* and *ASR* tasks can be considered out-of-domain.

Regarding the English to Spanish results of Table B.2, the UPC system shows also a high level of competitiveness, with scores close to those obtained by the best system. However, better ranking results are not observed for this translation direction when moving from *FTE* to *Verbatim* and *ASR*.

IBM stands for IBM (USA), ITC-irst for ITC-irst (Italy), RWTH for Aachen University (Germany), UED for University of Edinburgh (Scotland), UKA for University of Karlsruhe (Germany), UPC for Universitat Politecnica de Catalunya (Spain), DFKI for German Research Center for Artificial Intelligence (Germany), UW for University of Washington (USA), SYST for SYSTRAN (France) and LIMSI for LIMSI-CNRS (France).

Table B.2: *TC-Star'07 English-Spanish automatic (BLEU/NIST) comparative results for the three tasks (FTE, Verbatim and ASR). Site Rank is shown in parentheses for each measure. Euparl task.*

Site	<i>FTE</i>		<i>Verbatim</i>		<i>ASR</i>	
	BLEU	NIST	BLEU	NIST	BLEU	NIST
IBM	0.4754	9.890	0.4512 (3)	9.610	0.3577 (3)	8.620
ITC-irst	0.4981 (1)	10.23	0.4661 (1)	9.910	0.3597 (1)	8.750
RWTH	0.4944 (3)	10.16	0.4542 (2)	9.710	0.3591 (2)	8.720
UED	0.4950 (2)	10.11	-	-	-	-
UKA	0.4404	9.560	0.4010	9.080	0.3132	8.100
UPC	0.4885 (4)	10.06	0.4406 (4)	9.500	0.3476 (4)	8.560
DFKI	0.3632	8.700	-	-	-	-
UW	0.4850	10.01	0.4257	9.240	-	-
SYST	0.3629	8.570	0.3297	8.100	0.2393	7.030

B.2 IWSLT 2007

In October 2007, the C-STAR⁵ consortium organized the 3rd International Workshop on Spoken Language Translation (IWSLT'07), including an evaluation campaign whose details can be found in [For07].

The evaluation considered translation from Chinese and Arabic into English (*classical tasks*) consisting of translation of read speech in the travel domain, and from Italian and Chinese into English (*challenge tasks*), translation of spontaneous conversations in the travel domain. Up to 24 groups participated in the evaluation campaign that provided automatic and human evaluation measures.

UPC participated in the Chinese to English and Arabic to English tasks with a system performing *unfold* units, using POS-tag reordering rules and introducing *six* additional feature functions: a target language model, a word bonus, two lexicon models, a target tagged language model and a source tagged (reordered) language model. Although all publicly available data was allowed, we only used the provided data to train the system.

Following a similar approach to that in [Hab06], we used the *MADA+TOKAN* system for disambiguation and tokenization of the Arabic training/development/test sets. For disambiguation only diacritic uni-gram statistics were employed. For tokenization we used the D3 scheme with -TAGBIES option. The D3 scheme splits the following set of clitics: w+, f+, b+, k+, l+, Al+ and pronominal clitics. The -TAGBIES option produces Bies POS tags on all taggable tokens. Chinese preprocessing included re-segmentation using *ICTCLAS* [Zha03] and POS tagging using the freely available Stanford Parser⁶.

Comparative results are summarized in Tables B.3 for Arabic and B.4 for Chinese, which includes manual evaluation scores. Regarding the human evaluation (%Best), it consists of the average number of times that a system was judged to be better than any other system [CB07]. For each task, 300 sentences out of the 724 sentences in the evaluation set were randomly selected

⁵Consortium for Speech Translation Advanced Research, <http://www.c-star.org>

⁶<http://www-nlp.stanford.edu/software/lex-parser.shtml>

and presented to at least 3 evaluators. Since the ranking metric requires that each submission be compared to the other system outputs, each sentence may be presented multiple times but in the company of different sets of systems. Evaluators of each task and submission included 2 volunteers with experience in evaluating machine translation and 66 paid evaluators who were provided with a brief training in machine translation evaluation.

Table B.3: *IWSLT'07 Arabic-English human (%Better) and automatic (BLEU) comparative results for the two tasks (Clean and ASR). Site Rank is shown in parentheses for each measure.*

Site	Clean		ASR	
	%Better	BLEU	%Better	BLEU
DCU	45.1 (1)	0.4709	28.1	0.3942
UPC	42.9 (2)	0.4804 (3)	31.8 (1)	0.4445 (1)
UEKAE	36.4	0.4923 (1)	19.8	0.3679
UMD	36.0	0.4858 (2)	25.0	0.3908
UW	35.4	0.4161	26.9	0.4092
MIT	35.1	0.4553	31.4	0.4429
CMU	33.9	0.4463	25.5	0.3756
LIG	33.9	0.4135	24.2	0.3804
NTT	25.3	0.3403	25.5	0.3626
GREYC	21.7	0.3290	-	-
HKUST	13.1	0.1951	11.2	0.1420

Considering the Arabic-English pair, the UPC SMT system attains outstanding results, ranked in both cases (by human and automatic measures) as one of the best systems. Specially relevant is the performance achieved in the *ASR* task, where state-of-the-art results are obtained.

Notice that our system does not take multiple ASR output hypotheses into account but the single best. This gives additional relevance to the results achieved in the *ASR* task when compared to other systems.

As it can be seen, the UPC SMT system shows a fall in performance when considering the Chinese to English task. One of the reasons that can explain this situation is that our system seems to be less robust to noisy alignments (in special under scarce data availability) than standard phrase-based systems. The important reordering needs, the complexity of the Chinese vocabulary and the small data availability make the alignment process significantly more difficult in this translation task.

CASIA stands for Chinese Academy of Sciences, Institute of Automation, I2R for Institute for Infocomm Research (Singapore), ICT for Chinese Academy of Sciences, Inst. of Computing Technology (China) RWTH for Rheinisch-Westphalian Technical University (Germany), FBK for Fondazione Bruno Kesler (Italy), CMU for Carnegie Mellon University (USA), UPC for Technical University of Catalunya (Spain), XMU for Xiamen University (China), HKUST for University of Science and Technology (Hong Kong), MIT for Massachusetts Institute of Technology (USA), NTT for NTT Communication Science Laboratories (Japan), ATR for ATR Spoken Language Communication Research Laboratory (Japan), UMD for University of Maryland (USA), DCU for Dublin City University (Ireland), NUDT for National University of Defense Technology (China), LIG for University J. Fourier (France), MISTRAL for University of Montreal (Canada) and University of Avignon (France), GREYC for University of Caen (France) and UEDIN for

Table B.4: *IWSLT'07 Chinese-English human (%Better) and automatic (BLEU) comparative results for the Clean task. Site Rank is shown in parentheses for each measure.*

Site	<i>Clean</i>	
	%Better	BLEU
CASIA	37.6 (1)	0.3648 (5)
I2R	37.0 (2)	0.4077 (1)
ICT	34.8 (3)	0.3750 (2)
RWTH	32.4 (4)	0.3708 (4)
FBK	30.6 (5)	0.3472 (7)
CMU	30.6 (6)	0.3744 (3)
UPC	28.3 (7)	0.2991 (11)
XMU	28.1	0.2888
HKUST	25.5	0.3426 (8)
MIT	25.0	0.3631 (6)
NTT	24.6	0.2789
ATR	24.2	0.3133 (10)
UMD	23.6	0.3211
DCU	18.6	0.2737
NUDT	16.1	0.1934

University of Edinburgh (Scotland).

B.3 ACL 2007 WMT

In June 2007 took place the shared task of the 2007 ACL Workshop on Statistical Machine Translation. It is run on a one year basis beginning on 2005. This year, four language pairs were taken into account: Spanish-English, French-English, German-English and Czech-English, with translation tasks in both directions.

The shared task participants were provided with a common set of training and test data for all language pairs. The considered data was part of the European Parliament data set [Koe05b], and included also News Commentary data, which was the surprise out-of-domain test set of the previous year (News Commentary corpus). To lower the barrier of entrance to the competition, a complete baseline MT system, along with data resources was provided: sentence-aligned training corpora, development and dev-test sets, language models trained for each language, an open source decoder for phrase-based SMT (Moses [Koe07]), a training script to build models for mooses.

In addition to the Europarl test set, editorials from the Project Syndicate website⁷ were collected and employed as out-of-domain test.

The human evaluation was distributed across a number of people, including participants in the shared task, interested volunteers, and a small number of paid annotators. More than one hundred people participated out of which at least seventy five employed at least one hour of effort to account for three hundred thirty hours of total effort. Additional details of the shared

⁷<http://www.project-syndicate.com/>

Table B.5: WMT'07 Spanish-English human (*Adequacy/Fluency*) and automatic (*METEOR/BLEU*) comparative results for the two tasks (*Europarl* and *News*). Site Rank is shown in parentheses for each measure.

Site	Task	Adequacy	Fluency	METEOR	BLEU
cmu-syn	Europarl	0.552	0.568	0.602 (1)	0.323 (2)
cmu-uka		0.557	0.564	0.597	0.320
nrc		0.477	0.489	0.596	0.313
saar		0.328	0.336	0.542	0.245
systran		0.525	0.566	0.593	0.290
uedin		0.593 (1)	0.610 (1)	0.600 (2/3)	0.324 (1)
upc		0.587 (2)	0.604 (2)	0.600 (2/3)	0.322 (3)
upv	News	0.562	0.573	0.594	0.315
cmu-uka		0.522	0.495	0.640	0.299
nrc		0.479	0.464	0.641	0.299
saar		0.446	0.460	0.607	0.244
systran		0.525	0.503	0.628	0.259
uedin		0.546	0.534	0.661 (1)	0.327
upc		0.566 (1)	0.543 (1)	0.654 (2)	0.346 (1)
upv		0.435	0.459	0.638	0.283

task can be found in [CB07].

UPC participated in all tasks except for the Czech-English with a system performing SMR reordering using a set of automatically extracted word classes [Cj06] and introducing *four* additional feature functions: a target language model, a word bonus and two lexicon models. Further details in [Cj07b].

We used as preprocessing step the same Spanish morphology reduction employed for the system built for the third TC-Star evaluation, outlined in §B.1.

Tables B.5 and B.6 detail respectively the Spanish to English and English to Spanish results. Human (*Adequacy* and *Fluency*) and automatic (*METEOR* and *BLEU*) measures are used for both translation tasks (*Europarl* and *News*).

Considering the Spanish to English results, the UPC SMT system obtains very competitive results, specially for the out-of-domain task (*News*), where the human and automatic measures reward the system with the best results.

In the case of the English to Spanish results, in spite of achieving also highly competitive results the UPC system slightly loses performance in the comparison against other systems. The preprocessing step reducing the Spanish vocabulary seems to help more the Spanish to English direction than the English to Spanish one.

cmu-uka stands for Carnegie Mellon University (USA), University of Karlsruhe (Germany), cmu-syn for Carnegie Mellon University (USA), nrc for National Research Council (Canada), systran for SYSTRAN (France), uedin for University of Edinburgh (Scotland), upv for Technical University of Valencia (Spain), saar for Saarland University (Germany) and ucb for University of California Berkeley (USA).

Table B.6: WMT'07 English-Spanish human (Adequacy/Fluency) and automatic (METEOR/BLEU) comparative results for the two tasks (Europarl and News). Site Rank is shown in parentheses for each measure.

Site	Task	Adequacy	Fluency	METEOR	BLEU
cmu-uka	Europarl	0.563	0.581 (3)	0.333 (1)	0.311
nrc		0.546	0.548	0.322	0.299
systran		0.495	0.482	0.269	0.212
uedin		0.586 (1)	0.638 (1)	0.330 (2)	0.316 (1)
upc		0.584 (2)	0.578 (4)	0.327 (3)	0.312 (2)
upv		0.573	0.587 (2)	0.323	0.304
cmu-uka	News	0.510 (1/2)	0.492 (2)	0.368 (2)	0.327
nrc		0.408	0.392	0.362 (3)	0.311
systran		0.501	0.507 (1)	0.335	0.281
ucb		0.449	0.414	0.374 (1)	0.331 (1)
uedin		0.429	0.419	0.361 (4/5)	0.322
upc		0.510 (1/2)	0.488 (3)	0.361 (4/5)	0.328 (2)
upv		0.405	0.418	0.337	0.285

B.4 NIST 2006 MT Evaluation

The UPC SMT team participated by first time in the NIST Machine Translation evaluation on 2006.

The 2006 evaluation considered Arabic and Chinese the source languages under test, and English the target language. The text data consisted of newswire text documents, web-based newsgroup documents, human transcription of broadcast news, and human transcription of broadcast conversations. Performance was measured using BLEU. Human assessments were also taken into account on the evaluation, but only for the six best performing systems (in terms of BLEU).

Two evaluation data conditions were available for the participants: the (almost) *unlimited* data condition and the *large* data condition. The almost unlimited conditions has the single restriction of using data made available before February 2006. The large data conditions contemplates the use of data available from the LDC catalog.

UPC participated only on the large condition of both tasks (Chinese-English and Arabic-English). Unfortunately, we did not have enough time to prepare the evaluation before the test set release, what end up in a very poor data preprocessing of the huge amount of corpora available. The system was built performing *unfold* units, using heuristic constraints to allow for reordering (maximum distortion distance of 5 words and a limited number of 3 reordered words per sentence), and four additional models were employed: a target language model, a word bonus and two lexicon models.

Table B.7 shows the overall BLEU scores of both translation tasks. Results are sorted by the BLEU score of the NIST subset and reported separately for the GALE and the NIST subsets because they do not have the same number of reference translations. Fully detailed results can

be read in the NIST web site⁸.

Table B.7: *NIST’06 Arabic-English and Chinese-English comparative results (in terms of BLEU) for the two subsets (NIST and GALE) of the large data condition.*

<i>Arabic-English</i>			<i>Chinese-English</i>		
Site	NIST	GALE	Site	NIST	GALE
google	0.4281 (1)	0.1826	isi	0.3393 (1)	0.1413
ibm	0.3954	0.1674	google	0.3316	0.1470 (1)
isi	0.3908	0.1714	lw	0.3278	0.1299
rwth	0.3906	0.1639	rwth	0.3022	0.1187
apptek	0.3874	0.1918 (1)	ict	0.2913	0.1185
lw	0.3741	0.1594	edin	0.2830	0.1199
bbn	0.3690	0.1461	bbn	0.2781	0.1165
ntt	0.3680	0.1533	nrc	0.2762	0.1194
itcirst	0.3466	0.1475	itcirst	0.2749	0.1194
cmu-uka	0.3369	0.1392	umd-jhu	0.2704	0.1140
umd-jhu	0.3333	0.1370	ntt	0.2595	0.1116
edin	0.3303	0.1305	nict	0.2449	0.1106
sakhr	0.3296	0.1648	cmu	0.2348	0.1135
nict	0.2930	0.1192	msr	0.2314	0.0972
qmul	0.2896	0.1345	qmul	0.2276	0.0943
lcc	0.2778	0.1129	hkust	0.2080	0.0984
upc	0.2741 (17)	0.1149 (16)	upc	0.2071 (17)	0.0931 (17)
columbia	0.2465	0.0960	upenn	0.1958	0.0923
ucb	0.1978	0.0732	iscas	0.1816	0.0860
auc	0.1531	0.0635	lcc	0.1814	0.0813
dcu	0.0947	0.0320	xmu	0.1580	0.0747
kcs1	0.0522	0.0176	lingua	0.1341	0.0663
			kcs1	0.0512	0.0199
			ksu	0.0401	0.0218

As it can be seen, both tasks results are far from the best system’s results. While writing this document, our team is working on the NIST 2008 evaluation. Results will be easily accessible in the corresponding NIST web site.

apptek stands for Applications Technology Inc. (USA), auc for the American University in Cairo (Egypt), bbn for BBN Technologies (USA), cmu Carnegie Mellon University (USA), columbia for Columbia University (USA), dcu for Dublin City University (Ireland), google for Google (USA), hkust for Hong Kong University of Science and Technology (Hong Kong), ibm for IBM (USA), ict for Institute of Computing Technology Chinese Academy of Sciences (China), iscas for Institute of Software Chinese Academy of Sciences (China), isi for Information Sciences Institute (USA), itcirst for ITC-irst (Italy), ksu for Kansas State University(USA), kcs1 for KCSL Inc. (Canada), lw for Language Weaver (USA), lcc for Language Computer (USA), lingua for Lingua Technologies Inc. (Canada), msr for Microsoft Research (USA), nict for National Institute of Information and Communications Technology (Japan), ntt for NTT Communication Science Laboratories (Japan), nrc for National Research Council Canada (Canada), qmul for

⁸http://www.nist.gov/speech/tests/mt/doc/mt06eval_official_results.html

Queen Mary University of London (England), rwth for RWTH Aachen University (Germany), sakhr for Sakhr Software Co. (USA), ucb for University of California Berkeley (USA), edin for University of Edinburgh (Scotland), upenn for University of Pennsylvania (USA), upc for Universitat Politècnica de Catalunya (Spain), xmu for Xiamen University (China), cmu-uka for Carnegie Mellon University (USA), University of Karlsruhe (Germany), umd-jhu for University of Maryland, Johns Hopkins University (USA).

Appendix C

Publications by the author

The next is a list of major publications by the author:

1. **Improving SMT by coupling reordering and decoding.** Crego JM and Mariño JB. In Machine Translation, Volume 20, Number 3, pp 199-215, July 2007.
2. **Syntax-enhanced N-gram-based SMT.** Crego JM and Mariño JB. Proc. of the 11th Machine Translation Summit (MTsummitXI), pp 111-118 Copenhagen (Denmark), September 2007.
3. **Extending MARIE: an N-gram-based SMT decoder** Crego JM and Mariño JB. Proc. of the 45th annual meeting of the Association for Computational Linguistics (ACL'07/Poster), pp 213-216 Prague (Czech Republic), June 2007.
4. **Analysis and System Combination of Phrase- and N-gram-based Statistical Machine Translation Systems.** Costa-jussà MR, Crego JM, Vilar D, Fonollosa JAR, Mariño JB and Ney H. Proc. of the North American Chapter of the Association for Computational Linguistics, Human Language Technologies Conference (NAACL-HLT'07), pp 137-140 Rochester, NY (USA), April 2007.
5. **Discriminative Alignment Training without Annotated Data for Machine Translation.** Lambert P, Crego JM and Banchs R. Proc. of the North American Chapter of the Association for Computational Linguistics, Human Language Technologies Conference (NAACL-HLT'07), pp 85-88 Rochester, NY (USA), April 2007.
6. **N-gram-based Machine Translation.** Mariño JB, Banchs R, Crego JM, de Gispert A, Lambert P, Fonollosa JAR and Costa-jussà MR. In Computational Linguistics, Volume 32, Number 4, pp 527-549, December 2006.
7. **A Feasibility Study For Chinese-Spanish Statistical Machine Translation.** Banchs R, Crego JM, Lambert P and Mariño JB Proc. of the 5th Int. Symposium on Chinese Spoken Language Processing (ISCSLP'06), pp 681-692 Kent Ridge (Singapore), December 2006.
8. **Reordering Experiments for N-gram-based SMT.** Crego JM and Mariño JB. Ist IEEE/ACL International Workshop on Spoken Language Technology (SLT'06), pp 242-245 Palm Beach (Aruba), December 2006.

9. **Integration of POSTag-based source reordering into SMT decoding by an extended search graph.** Crego JM and Mariño JB. 7th biennial conference of the Association for Machine Translation in the Americas (AMTA'06), pp 29-36 Boston (USA), August 2006.
10. **Integración de reordenamientos en el algoritmo de decodificación en traducción automática estocástica.** Crego JM and Mariño JB. Procesamiento del Lenguaje Natural, núm 6 (SEPLN'06) Zaragoza (Spain), September 2006.
11. **The TALP Ngram-based SMT System for IWSLT'05.** Crego JM, Mariño JB and de Gispert A. Proc. of the 2nd Int. Workshop on Spoken Language Translation (IWSLT'05), pp 191-198 Pittsburgh (USA), October 2005.
12. **Ngram-based versus Phrase-based Statistical Machine Translation.** Crego JM, Costa-jussà MR, Mariño JB and Fonollosa JAR. Proc. of the 2nd Int. Workshop on Spoken Language Translation (IWSLT'05), pp 177-184 Pittsburgh (USA), October 2005.
13. **Reordered search and Tuple Unfolding for Ngram-based SMT.** Crego JM, Mariño JB and de Gispert A. Proc. of the 10th Machine Translation Summit (MTsummitX), pp 283-289 Phuket (Thailand), September 2005.
14. **An Ngram-based Statistical Machine Translation Decoder.** Crego JM, Mariño JB and de Gispert A. Proc. of the 9th European Conf. on Speech Communication and Technology (Interspeech'05), pp 3185-3188 Lisbon (Portugal), September 2005.
15. **Improving Statistical Machine Translation by Classifying and Generalizing Inflected Verb Forms.** de Gispert A, Mariño JB and Crego JM. Proc. of the 9th European Conf. on Speech Communication and Technology, (Interspeech'05), pp 3193-3196 Lisbon (Portugal), September 2005.
16. **Algoritmo de Decodificación de Traducción Automática Estocástica basado en N-gramas.** Crego JM, Mariño JB and de Gispert A. Procesamiento del Lenguaje Natural, núm 5 (SEPLN'05), pp 82-95 Granada (Spain), September 2005.
17. **Clasificación y generalización de formas verbales en sistemas de traducción estocástica.** de Gispert A, Mariño JB and Crego JM. Procesamiento del Lenguaje Natural, núm 5 (SEPLN'05), pp 335-342 Granada (Spain), September 2005.
18. **Finite-state-based and Phrase-based Statistical Machine Translation.** Crego JM, Mariño JB and de Gispert A. Proc. of the 8th Int. Conf. on Spoken Language Processing (ICSLP'04), pp 37-40 Jeju island (Korea), October 2004.
19. **Phrase-based Alignment combining corpus cooccurrences and linguistic knowledge.** de Gispert A, Mariño JB and Crego JM. Proc. of the Int. Workshop on Spoken Language Translation (IWSLT'04), pp 85-90 Kyoto (Japan), October 2004.

Publications in review process:

1. **Using Shallow Syntax Information to Improve Word Alignment and Reordering in SMT.** Crego JM., Habash N. and Mariño JB Submitted to Proc. of the 46th annual meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'08), Ohio (USA), June 2008.

2. **Decoding *N*-gram-based translation models.** Crego JM and Mariño JB. Submitted to Machine Translation.
3. **A Linguistically-motivated Reordering Framework for SMT.** Crego JM and Mariño JB. Submitted to Computational Linguistics.

Other publications:

1. **The TALP Ngram-based SMT System for IWSLT 2007.** Lambert P, Costa-jussà MR, Crego JM, Khalilov M, Mariño JB, Banchs R, Fonollosa JAR and Schwenk H. Proc. of the 4th Int. Workshop on Spoken Language Translation (IWSLT'07), pp Trento (Italy), October 2007.
2. **Ngram-based system enhanced with multiple weighted reordering hypotheses.** Costa-jussà MR, Lambert P, Crego JM, Khalilov M, Fonollosa JAR, Mariño JB and Banchs R. Proc. of the Association for Computational Linguistics, Second Workshop on Statistical Machine Translation (ACL'07/Wkshp), pp 167-170 Prague (Czech Republic), June 2007.
3. **The TALP Ngram-based SMT System for IWSLT 2006.** Crego JM, de Gispert A, Lambert P, Khalilov M, Costa-jussà MR, Mariño JB, Banchs R and Fonollosa JAR. Proc. of the 3rd Int. Workshop on Spoken Language Translation (IWSLT'06), pp 116-122 Kyoto (Japan), November 2006.
4. **TALP Phrase-based System and TALP System Combination for the IWSLT 2006.** Costa-jussà MR, Crego JM, de Gispert A, Lambert P, Khalilov M, Fonollosa JAR, Mariño JB and Banchs R. Proc. of the 3rd Int. Workshop on Spoken Language Translation (IWSLT'06), pp 123-129 Kyoto (Japan), November 2006.
5. **UPC's Bilingual N-gram Translation System.** Mariño JB, Banchs R, Crego JM, de Gispert A, Lambert P, Fonollosa JAR, Costa-jussà MR and Khalilov M. TC-Star Speech to Speech Translation Workshop (TC-Star'06/Wkshp), pp 43-48 Barcelona (Spain), June 2006.
6. **N-gram-based SMT System Enhanced with Reordering Patterns.** Crego JM, de Gispert A, Lambert P, Costa-jussà MR, Khalilov M, Banchs R, Mariño JB and Fonollosa JAR. Proc. of the HLT-NAACL Workshop on Statistical Machine Translation (HLT-NAACL'06/Wkshp), pp 162-165 New York (USA), June 2006.
7. **TALP Phrase-based statistical translation system for European language pairs.** Costa-jussà MR, Crego JM, de Gispert A, Lambert P, Khalilov M, Banchs R, Mariño JB and Fonollosa JAR. Proc. of the HLT-NAACL Workshop on Statistical Machine Translation (HLT-NAACL'06/Wkshp), pp 142-145 New York (USA), June 2006.
8. **Bilingual N-gram Statistical Machine Translation.** Mariño JB, Banchs R, Crego JM, de Gispert A, Lambert P, Fonollosa JAR and Costa-jussà MR. Proc. of the 10th Machine Translation Summit (MTsummitX), pp 275-282 Phuket (Thailand), September 2005.
9. **Modelo estocástico de traducción basado en N-gramas de tuplas bilingües y combinación log-lineal de características.** Mariño JB, Crego JM, Lambert P, Banchs R, de Gispert A, Fonollosa JAR and Costa-jussà MR. Procesamiento del Lenguaje Natural, núm 5 (SEPLN'05), pp 69-76 Granada (Spain), September 2005.

10. **Statistical Machine Translation of Euparl Data by using Bilingual N-grams.** Banchs R, Crego JM, de Gispert A, Lambert P and Mariño JB Proc. of the ACL Workshop on Building and Using Parallel Texts (ACL'05/Wkshp), pp 133-136 Ann Arbor (USA), June 2005.
11. **Bilingual connections for Trilingual Corpora: An XML approach.** Arranz V, Castell N, Crego JM, Gimenez J, de Gispert A and Lambert P, In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04), pp 1459-1462 Lisbon (Portugal), May 2004.
12. **Els sistemes de reconeixement de veu i traducció automàtica en català: present i futur.** Anguera X, Anguita J, Farrús M, Crego JM, de Gispert A, Hernando X and Nadeu C. 2on Congrés d'Enginyeria en Llengua Catalana (CELC'04) Andorra (Andorra), November 2004.

Bibliography

- [A.72] Aho A., and Ullman J., “The theory of parsing, translation and compiling, volume i: Parsing”, 1972.
- [Aki04] Y Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, and J. Tsujii, “Overview of the iwslt04 evaluation campaign”, *Proc. of the 1st Int. Workshop on Spoken Language Translation, IWSLT’04*, pages. 1–12, October 2004.
- [Als96] H. Alshawi, “Head automata for speech translation”, *Proc. of the 4th Int. Conf. on Spoken Language Processing, ICSLP’96*, pages. 2360–2364, October 1996.
- [AO99] Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, D. Melamed, F.J. Och, D. Purdy, N.A. Smith, and D. Yarowsky, “Statistical machine translation: Final report”, Tech. rep., Johns Hopkins University Summer Workshop, Baltimore, MD, USA, 1999.
- [AO06] Yaser Al-Onaizan, and Kishore Papineni, “Distortion models for statistical machine translation”, *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages. 529–536, Association for Computational Linguistics, Sydney, Australia, July 2006.
- [Arn95] D. Arnold, and L. Balkan, “Machine translation: an introductory guide”, *Comput. Linguist.*, Vol. 21, n^o 4, pages. 577–578, 1995.
- [Aru06] A. Arun, A. Axelrod, Birch A., Callison-Burch C., H. Hoang, P. Koehn, M. Osborne, and D. Talbot, “Edinburgh system description for the 2006 tcstar spoken language translation evaluation”, *TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, June 2006.
- [B.94] Bonnie B., “Machine translation divergences: a formal description and proposed solution”, *Computational Linguistics*, Vol. 20, n^o 4, pages. 597–633, 1994.
- [Bab04] B. Babych, and T. Hartley, “Extending the bleu mt evaluation method with frequency weightings”, *42nd Annual Meeting of the Association for Computational Linguistics*, pages. 621–628, July 2004.
- [Ban99] S. Bangalore, and A. Joshi, “Supertagging: An approach to almost parsing”, *Computational Linguistics*, Vol. 25, n^o 2, pages. 237–265, 1999.
- [Ban00a] S. Bangalore, and G. Riccardi, “Finite-state models for lexical reordering in spoken language translation”, *Proc. of the 6th Int. Conf. on Spoken Language Processing, ICSLP’00*, October 2000.

- [Ban00b] S. Bangalore, and G. Riccardi, “Stochastic finite-state models for spoken language machine translation”, *Proc. Workshop on Embedded Machine Translation Systems*, pages. 52–59, April 2000.
- [Ban05] S. Banerjee, and A. Lavie, “METEOR: An automatic metric for mt evaluation with improved correlation with human judgments”, *Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages. 65–72, June 2005.
- [Ber94] A. Berger, P. Brown, S. Della Pietra, V. Della Pietra, and J. Gillet, “The candide system for machine translation”, *Proceedings of the Arpa Workshop on Human Language Technology*, March 1994.
- [Ber96] A. Berger, S. Della Pietra, and V. Della Pietra, “A maximum entropy approach to natural language processing”, *Computational Linguistics*, Vol. 22, n° 1, pages. 39–72, March 1996.
- [Ber05] N. Bertoldi, and M. Federico, “A new decoder for spoken language translation based on confusion networks”, *IEEE Automatic Speech Recognition and Understanding Workshp, ASRU’05*, December 2005.
- [Ber06] N. Bertoldi, R. Cattoni, M. Cettolo, B. Chen, and M. Federico, “Itc-first at the 2006 tc-star slt evaluation campaign”, *TC-STAR Workshop on Speech-to-Speech Translation*, pages. 19–24, Barcelona, Spain, June 2006.
- [Ber07] N. Bertoldi, R. Zens, and M. Federico, “Speech translation by confusion network decoding”, *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP’07)*, April 2007.
- [BH60] Y. Bar-Hillel, “The present state of automatic translation of languages”, *Advances in Computers*, Vol. 1, pages. 91–163, 1960.
- [Bla04] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing, “Confidence estimation for machine translation”, *Proc. of the 20th Int. Conf. on Computational Linguistics, COLING’04*, pages. 315–321, August 2004.
- [Bra00] T. Brants, “TnT – a statistical part-of-speech tagger”, *Proc. of the Sixth Applied Natural Language Processing (ANLP-2000)*, Seattle, WA, 2000.
- [Bro90] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J.D. Lafferty, R. Mercer, and P.S. Roossin, “A statistical approach to machine translation”, *Computational Linguistics*, Vol. 16, n° 2, pages. 79–85, 1990.
- [Bro93] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer, “The mathematics of statistical machine translation: Parameter estimation”, *Computational Linguistics*, Vol. 19, n° 2, pages. 263–311, 1993.
- [Buc04] Tim Buckwalter, “Issues in arabic orthography and morphology analysis”, Ali Farghaly, Karine Megerdooimian (eds.), *COLING 2004 Computational Approaches to Arabic Script-based Languages*, pages. 31–34, COLING, Geneva, Switzerland, August 28th 2004.

- [Car04] X. Carreras, I. Chao, L. Padró, and M. Padró, “Freeling: An open-source suite of language analyzers”, *4th Int. Conf. on Language Resources and Evaluation, LREC’04*, May 2004.
- [Cas01] F. Casacuberta, “Finite-state transducers for speech-input translation”, *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU’01*, December 2001.
- [Cas04] F. Casacuberta, and E. Vidal, “Machine translation with inferred stochastic finite-state transducers”, *CL*, Vol. 30, nº 2, pags. 205–225, 2004.
- [CB06] Ch. Callison-Burch, M. Osborne, and Ph. Koehn, “Re-evaluating the role of bleu in machine translation research”, *13th Conf. of the European Chapter of the Association for Computational Linguistics*, pags. 249–246, April 2006.
- [CB07] C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder, “(meta-) evaluation of machine translation”, *Proceedings of the Second Workshop on Statistical Machine Translation*, pags. 136–158, Association for Computational Linguistics, Prague, Czech Republic, June 2007.
- [Chi05] D. Chiang, “A hierarchical phrase-based model for statistical machine translation”, *43rd Annual Meeting of the Association for Computational Linguistics*, pags. 263–270, June 2005.
- [Cj06] M.R. Costa-jussà, and J.A.R. Fonollosa, “Statistical machine reordering”, *Proc. of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP’06*, July 2006.
- [Cj07a] Marta R. Costa-jussà, Josep M. Crego, José B. Mariño, David Vilar, and Hermann Ney, “Analysis and System Combination of Phrase- and *N*-gram-based Statistical Machine Translation Systems”, *submitted to HLT-NAACL’07*, 2007.
- [Cj07b] M.R. Costa-jussà, J.M. Crego, P. Lambert, Khalilov, J.B. Mariño, J.A.R. Fonollosa, M., and R. Banchs, “*N*-gram-based statistical machine translation enhanced with weighted reordering hypotheses”, *Proceedings of the Second Workshop on Statistical Machine Translation*, pags. 167–170, Association for Computational Linguistics, Prague, Czech Republic, June 2007.
- [Col99] M. Collins, *Head-driven Statistical Models for Natural Language Parsing*, PhD Thesis, University of Pennsylvania, 1999.
- [Col05a] M. Collins, Ph. Koehn, and I. Kucerova, “Clause restructuring for statistical machine translation”, *43rd Annual Meeting of the Association for Computational Linguistics*, pags. 531–540, June 2005.
- [Col05b] Michael Collins, Philipp Koehn, and Ivona Kucerova, “Clause restructuring for statistical machine translation”, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pags. 531–540, Association for Computational Linguistics, Ann Arbor, Michigan, June 2005.
- [Cre] J.M. Crego, and J.B. Mariño, “Improving statistical mt by coupling reordering and decoding”, .

- [Cre04] J.M. Crego, J.B. Mariño, and A. de Gispert, “Finite-state-based and phrase-based statistical machine translation”, *Proc. of the 8th Int. Conf. on Spoken Language Processing, ICSLP’04*, pages. 37–40, October 2004.
- [Cre05a] J.M. Crego, A. de Gispert, and J.B. Mariño, “TALP: The UPC tuple-based SMT system”, *Proc. of the 2nd Int. Workshop on Spoken Language Translation, IWSLT’05*, pages. 191–198, October 2005.
- [Cre05b] J.M. Crego, J.B. Mariño, and A. de Gispert, “An ngram-based statistical machine translation decoder”, *Proc. of the 9th European Conference on Speech Communication and Technology, Interspeech’05*, pages. 3193–3196, September 2005.
- [Cre05c] J.M. Crego, J.B. Mariño, and A. de Gispert, “Reordered search and tuple unfolding for ngram-based smt”, *Proc. of the MT Summit X*, pages. 283–89, September 2005.
- [Cre06a] J.M. Crego, and J.B. Mariño, “Integration of postag-based source reordering into smt decoding by an extended search graph”, *Proc. of the 7th Conf. of the Association for Machine Translation in the Americas*, pages. 29–36, August 2006.
- [Cre06b] J.M. Crego, and J.B. Mariño, “Reordering experiments for n-gram-based smt”, *1st IEEE/ACL Workshop on Spoken Language Technology*, December 2006.
- [Cre07a] J.M. Crego, and J.B. Mariño, “Extending marie: an n-gram-based smt decoder”, *45rd Annual Meeting of the Association for Computational Linguistics*, April 2007.
- [Cre07b] J.M. Crego, and J.B. Mariño, “Syntax-enhanced n-gram-based smt”, *Proc. of the MT Summit XI*, September 2007.
- [Dia04] Mona Diab, Kadri Hacioglu, and Daniel Jurafsky, “Automatic tagging of arabic text: From raw text to base phrase chunks”, Daniel Marcu Susan Dumais, Salim Roukos (eds.), *HLT-NAACL 2004: Short Papers*, pages. 149–152, Association for Computational Linguistics, Boston, Massachusetts, USA, May 2 - May 7 2004.
- [Din05] Yuan Ding, and Martha Palmer, “Machine translation using probabilistic synchronous dependency insertion grammars”, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages. 541–548, Association for Computational Linguistics, Ann Arbor, Michigan, June 2005.
- [Dod02] G. Doddington, “Automatic evaluation of machine translation quality using n-gram co-occurrence statistics”, *Proc. ARPA Workshop on Human Language Technology*, 2002.
- [Dor94] B.J. Dorr, “Machine translation: a view from the lexicon”, *Comput. Linguist.*, Vol. 20, n° 4, pages. 670–676, 1994.
- [E.61] Irons E., “A syntax-directed compiler for algol 60”, *ACM*, Vol. 4, n° 1, pages. 51–55, 1961.
- [Eck05] M. Eck, and Ch. Hori, “Overview of the IWSLT 2005 Evaluation Campaign”, *Proc. of the 2nd Int. Workshop on Spoken Language Translation, IWSLT’05*, pages. 11–32, October 2005.

- [Eis03] J. Eisner, “Learning non-isomorphic tree mappings for machine translation”, *ACL03*, pags. 205–208, Association for Computational Linguistics, Morristown, NJ, USA, 2003.
- [For07] C. Fordyce, “Overview of the IWSLT 2007 Evaluation Campaign”, *IWSLT07*, pags. 1–12, Trento, Italy, 2007.
- [Gal04] M. Galley, and Hopkins M., “What’s in a translation rule?”, *HLTNAACL04*, pags. 273–280, Boston, MA, May 2004.
- [Ger01] U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada, “Fast decoding and optimal decoding for machine translation”, *39th Annual Meeting of the Association for Computational Linguistics*, pags. 228–235, July 2001.
- [Ger03] U. Germann, “Greedy decoding for statistical machine translation in almost linear time”, *Proc. of the Human Language Technology Conference, HLT-NAACL’2003*, May 2003.
- [Gil03] D. Gildea, “Loosely tree-based alignment for machine translation”, *ACL03*, pags. 80–87, Sapporo, Japan, July 2003.
- [Gim06] J. Giménez, and E. Amigó, “Iqmt: A framework for automatic machine translation evaluation”, *5th Int. Conf. on Language Resources and Evaluation, LREC’06*, pags. 22–28, May 2006.
- [Gis04] A. de Gispert, and J.B. Mariño, “TALP: Xgram-based Spoken Language Translation System”, *Proc. of the 1st Int. Workshop on Spoken Language Translation, IWSLT’04*, pags. 85–90, October 2004.
- [Gis06] A. de Gispert, and J.B. Mariño, “Linguistic tuple segmentation in ngram-based statistical machine translation”, *Proc. of the 9th Int. Conf. on Spoken Language Processing, ICSLP’06*, pags. 1149–1152, September 2006.
- [Gra04] J. Graehl, and K. Knight, “Training tree transducers”, *HLTNAACL04*, pags. 105–112, Association for Computational Linguistics, Boston, Massachusetts, USA, May 2 - May 7 2004.
- [GV03] I. García Varea, *Traducción automática estadística: modelos de traducción basados en máxima entropía y algoritmos de búsqueda*, PhD Thesis in Informatics, Dep. de Sistemes Informàtics i Computació, Universitat Politècnica de València, 2003.
- [Hab05] N. Habash, and O. Rambow, “Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop”, *43rd Annual Meeting of the Association for Computational Linguistics*, pags. 573–580, Association for Computational Linguistics, Ann Arbor, MI, June 2005.
- [Hab06] N. Habash, and F. Sadat, “Arabic preprocessing schemes for statistical machine translation”, *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pags. 49–52, Association for Computational Linguistics, New York City, USA, June 2006.
- [Hab07] N. Habash, “Syntactic preprocessing for statistical machine translation”, *Proc. of the MT Summit XI*, September 2007.

- [Has06] H. Hassan, M. Hearne, A. Way, and K. Sima'an, "Syntactic phrase-based statistical machine translation", *1st IEEE/ACL Workshop on Spoken Language Technology*, December 2006.
- [Has07] H. Hassan, K. Sima'an, and A. Way, "Supertagged phrase-based statistical machine translation", *ACL07*, pages. 288–295, Prague, Czech Republic, June 2007.
- [Hew05] S. Hewavitharana, B. Zhao, A.S. Hildebrand, M. Eck, Ch. Hori, S. Vogel, and A. Waibel, "The cmu statistical machine translation system for IWSLT 2005", *Proc. of the 2nd Int. Workshop on Spoken Language Translation, IWSLT'05*, pages. 63–70, October 2005.
- [Hua06] Liang Huang, Kevin Knight, and Aravind Joshi, "A syntax-directed translator with extended domain of locality", *Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages. 1–8, Association for Computational Linguistics, New York City, New York, June 2006.
- [Hut92] W.J. Hutchins, and H.L. Somers, "An introduction to machine translation", 1992.
- [Kan05] S. Kanthak, D. Vilar, E. Matusov, R. Zens, and H. Ney, "Novel reordering approaches in phrase-based statistical machine translation", *Proc. of the ACL Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pages. 167–174, June 2005.
- [Kir06] K. Kirchhoff, M. Yang, and K. Duh, "Statistical machine translation of parliamentary proceedings using morpho-syntactic knowledge", *TC-STAR Workshop on Speech-to-Speech Translation*, pages. 57–62, Barcelona, Spain, June 2006.
- [Kne95] R. Kneser, and H. Ney, "Improved backing-off for m-gram language modeling", *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'95)*, Vol. 1, pages. 181–184, 1995.
- [Kni98] K. Knight, and Y. Al-Onaizan, "Translation with finite-state devices", *Proc. of the 3rd Conf. of the Association for Machine Translation in the Americas*, pages. 421–437, October 1998.
- [Kni99] K. Knight, "Decoding complexity in word replacement translation models", *Computational Linguistics*, Vol. 26, n^o 2, pages. 607–615, 1999.
- [Koe03a] K. Koehn, and Knight K., "Empirical methods for compound splitting", *10th Conf. of the European Chapter of the Association for Computational Linguistics*, pages. 347–354, April 2003.
- [Koe03b] Ph. Koehn, F.J. Och, and D. Marcu, "Statistical phrase-based translation", *Proc. of the Human Language Technology Conference, HLT-NAACL'2003*, May 2003.
- [Koe04] Ph. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models", *Proc. of the 6th Conf. of the Association for Machine Translation in the Americas*, pages. 115–124, October 2004.
- [Koe05a] Axelrod A. Birch Mayne Callison-Burch Osborne M. Koehn, P., and D. Talbot, "Edinburgh system description for the 2005 iwslt speech translation evaluation", pages. 63–70, Pittsburgh, USA, October 2005.

- [Koe05b] Ph. Koehn, “Europarl: A parallel corpus for statistical machine translation”, *Proc. of the MT Summit X*, pages. 79–86, September 2005.
- [Koe05c] Ph. Koehn, and C. Monz, “Shared task: Statistical Machine Translation between European Languages”, *Proc. of the ACL Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pages. 119–124, June 2005.
- [Koe06] Ph. Koehn, and C. Monz, “Manual and automatic evaluation of machine translation between european languages”, *Proceedings of the Workshop on Statistical Machine Translation*, pages. 102–21, Association for Computational Linguistics, New York City, June 2006.
- [Koe07] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst, “Moses: Open source toolkit for statistical machine translation”, *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages. 177–180, Association for Computational Linguistics, Prague, Czech Republic, June 2007.
- [Kuh06] R. Kuhn, G. Foster, S. Larkin, and N. Ueffing, “Portage phrase-based system for chinese-to-english translation”, *TC-STAR Workshop on Speech-to-Speech Translation*, pages. 75–80, Barcelona, Spain, June 2006.
- [Kum04] S. Kumar, and W. Byrne, “Minimum bayes-risk decoding for statistical machine translation”, *Proc. of the Human Language Technology Conference, HLT-NAACL’2004*, pages. 169–176, May 2004.
- [Kum05] S. Kumar, and W. Byrne, “Local phrase reordering models for statistical machine translation”, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages. 161–168, Association for Computational Linguistics, Vancouver, British Columbia, Canada, October 2005.
- [Lan06] P. Langlais, and F. Gotti, “Phrase-based smt with shallow tree-phrases”, *Proceedings of the Workshop on Statistical Machine Translation*, pages. 39–46, June 2006.
- [Lee06] Y.S. Lee, Y. Al-Onaizan, K. Papineni, and S. Roukos, “Ibm spoken language translation system”, *TC-STAR Workshop on Speech-to-Speech Translation*, pages. 13–18, Barcelona, Spain, June 2006.
- [Lin04a] Chin-Yew Lin, “ROUGE: a package for automatic evaluation of summaries”, *ACL 2004 Workshop: Text Summarization Branches Out*, Barcelona, Spain, July 2004.
- [Lin04b] Chin-Yew Lin, and F.J. Och, “ORANGE: a method for evaluating automatic evaluation metrics for machine translation”, *Proc. of the 20th Int. Conf. on Computational Linguistics, COLING’04*, pages. 501–507, August 2004.
- [Liu06] Yang (1) Liu, Qun Liu, and Shouxun Lin, “Tree-to-string alignment template for statistical machine translation”, *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages. 609–616, Association for Computational Linguistics, Sydney, Australia, July 2006.

- [Maa04] Mohamed Maamouri, and Ann Bies, “Developing an arabic treebank: Methods, guidelines, procedures, and tools”, Ali Farghaly, Karine Megerdooian (eds.), *COLING 2004 Computational Approaches to Arabic Script-based Languages*, pages. 2–9, COLING, Geneva, Switzerland, August 28th 2004.
- [Mar02] D. Marcu, and W. Wong, “A phrase-based, joint probability model for statistical machine translation”, *Proc. of the Conf. on Empirical Methods in Natural Language Processing, EMNLP’02*, pages. 133–139, July 2002.
- [Mar06] D. Marcu, Wong. W, A. Echiabi, and K. Knight, “Spm: Statistical machine translation with syntactified target language phrases”, *Proc. of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP’06*, pages. 44–52, Sydney, Australia, July 2006.
- [Mat06] E. Matusov, R. Zens, D. Vilar, A. Mauser, M. Popovic, S. Hasan, and H. Ney, “The rwth machine translation system”, *TC-STAR Workshop on Speech-to-Speech Translation*, pages. 31–36, Barcelona, Spain, June 2006.
- [Mel03] D. Melamed, “Multitext grammars and synchronous parsers”, *NAACL03*, pages. 79–86, Edmonton, Canada, 2003.
- [Mel04] D. Melamed, “Statistical machine translation by parsing”, *42nd Annual Meeting of the Association for Computational Linguistics*, pages. 653–661, July 2004.
- [Nag06] Masaaki Nagata, Kuniko Saito, Kazuhide Yamamoto, and Kazuteru Ohashi, “A clustered global phrase reordering model for statistical machine translation”, *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages. 713–720, Association for Computational Linguistics, Sydney, Australia, July 2006.
- [Nel65] J.A. Nelder, and R. Mead, “A simplex method for function minimization”, *The Computer Journal*, Vol. 7, pages. 308–313, 1965.
- [Nie01] S. Nießen, and H. Ney, “Morpho-syntactic analysis for reordering in statistical machine translation”, pages. 247–252, September 2001.
- [Nie04] S. Nießen, and H. Ney, “Statistical machine translation with scarce resources using morpho-syntactic information”, *Computational Linguistics*, Vol. 30, n^o 2, pages. 181–204, June 2004.
- [Och99] F.J. Och, Ch. Tillmann, and H. Ney, “Improved alignment models for statistical machine translation”, *Proc. of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora*, pages. 20–28, June 1999.
- [Och00a] F.J. Och, and H. Ney, “A comparison of alignment models for statistical machine translation”, *Proc. of the 18th Int. Conf. on Computational Linguistics, COLING’00*, pages. 1086–1090, July 2000.
- [Och00b] F.J. Och, and H. Ney, “Improved statistical alignment models”, *38th Annual Meeting of the Association for Computational Linguistics*, pages. 440–447, October 2000.

- [Och01] F.J. Och, N. Ueffing, and H. Ney, “An efficient A* search algorithm for statistical machine translation”, *Data-Driven Machine Translation Workshop, 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages. 55–62, July 2001.
- [Och02] F.J. Och, and H. Ney, “Discriminative training and maximum entropy models for statistical machine translation”, *40th Annual Meeting of the Association for Computational Linguistics*, pages. 295–302, July 2002.
- [Och03a] F.J. Och, “Giza++ software. <http://www-i6.informatik.rwth-aachen.de/~och/software/giza++.html>”, Tech. rep., RWTH Aachen University, 2003.
- [Och03b] F.J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev, “Syntax for statistical machine translation”, Tech. Rep. Summer Workshop Final Report, Johns Hopkins University, Baltimore, USA, 2003.
- [Och03c] F.J. Och, and H. Ney, “A systematic comparison of various statistical alignment models”, *Computational Linguistics*, Vol. 29, n^o 1, pages. 19–51, March 2003.
- [Och04a] F.J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev, “A smorgasbord of features for statistical machine translation”, *Proc. of the Human Language Technology Conference, HLT-NAACL’2004*, pages. 161–168, May 2004.
- [Och04b] F.J. Och, and H. Ney, “The alignment template approach to statistical machine translation”, *Computational Linguistics*, Vol. 30, n^o 4, pages. 417–449, December 2004.
- [Olt06] M. Olteanu, Ch. Davis, I. Volosen, and D. Moldovan, “Phramer - an open source statistical phrase-based translator”, *Proceedings on the Workshop on Statistical Machine Translation*, pages. 146–149, Association for Computational Linguistics, New York City, June 2006.
- [Ort05] D. Ortiz, I. García-Varea, and F. Casacuberta, “Thot: a toolkit to train phrase-based statistical translation models”, *Proc. of the MT Summit X*, pages. 141–148, September 2005.
- [P.68] Lewis P., and Stearns R., “Syntax-directed transduction”, *ACM*, Vol. 15, n^o 3, pages. 465–488, 1968.
- [Pap98] K.A. Papineni, S. Roukos, and R.T. Ward, “Maximum likelihood and discriminative training of direct translation models”, *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing*, pages. 189–192, May 1998.
- [Pap01] K. Papineni, S. Roukos, T. Ward, and W. Zhu, “Bleu: a method for automatic evaluation of machine translation”, Tech. Rep. RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, 2001.
- [Pap02] K. Papineni, S. Roukos, T. Ward, and W-J. Zhu, “Bleu: A method for automatic evaluation of machine translation”, *40th Annual Meeting of the Association for Computational Linguistics*, pages. 311–318, July 2002.

- [Pat06] A. Patry, F. Gotti, and P. Langlais, “Mood at work: Ramses versus pharaoh”, *Proceedings on the Workshop on Statistical Machine Translation*, pages. 126–129, Association for Computational Linguistics, New York City, June 2006.
- [Pop06a] M. Popovic, A. de Gispert, D. Gupta, P. Lambert, H. Ney, J.B. Mariño, M. Federico, and R. Banchs, “Morpho-syntactic information for automatic error analysis of statistical machine translation output”, *Proceedings of the Workshop on Statistical Machine Translation*, pages. 1–6, Association for Computational Linguistics, New York City, June 2006.
- [Pop06b] M. Popovic, and H. Ney, “Error analysis of verb inflections in spanish translation output”, *TC-STAR Workshop on Speech-to-Speech Translation*, pages. 99–103, Barcelona, Spain, June 2006.
- [Pop06c] M. Popovic, and H. Ney, “Pos-based word reorderings for statistical machine translation”, *5th Int. Conf. on Language Resources and Evaluation, LREC’06*, pages. 1278–1283, May 2006.
- [Prz06] M. Przybocki, G. Sanders, and A. Le, “Edit distance: A metric for machine translation evaluation”, *5th Int. Conf. on Language Resources and Evaluation, LREC’06*, pages. 2038–2043, May 2006.
- [Qua05] V.H. Quan, M. Federico, and M. Cettolo, “Integrated n-best re-ranking for spoken language translation”, *Proc. of the 9th European Conference on Speech Communication and Technology, Interspeech’05*, September 2005.
- [Qui05] Ch. Quirk, A. Menezes, and C. Cherry, “Dependency treelet translation: Syntactically informed phrasal SMT”, *43rd Annual Meeting of the Association for Computational Linguistics*, pages. 271–279, June 2005.
- [Sha49a] C.E. Shannon, “Communication theory of secrecy systems”, *The Bell System Technical Journal*, Vol. 28, pages. 656–715, 1949.
- [Sha49b] C.E. Shannon, and W. Weaver, *The mathematical theory of communication*, University of Illinois Press, Urbana, IL, 1949.
- [Sha51] C.E. Shannon, “Prediction and entropy of printed english”, *The Bell System Technical Journal*, Vol. 30, pages. 50–64, 1951.
- [She04] L. Shen, A. Sarkar, and F.J. Och, “Discriminative reranking for machine translation”, Daniel Marcu Susan Dumais, Salim Roukos (eds.), *Proc. of the Human Language Technology Conference, HLT-NAACL’2004*, pages. 177–184, Association for Computational Linguistics, Boston, Massachusetts, USA, May 2004.
- [Shi90] S. Shieber, and Y. Schabes, “Synchronous tree-adjointing grammars”, *Proceedings of the 13th conference on Computational linguistics*, pages. 253–258, Association for Computational Linguistics, Morristown, NJ, USA, 1990.
- [Sno05] M. Snover, B. Dorr, R. Schwartz, J. Makhoul, L. Micciula, and R. Weischedel, “A study of translation error rate with targeted human annotation”, Tech. Rep. LAMP-TR-126, CS-TR-4755, UMIACS-TR-2005-58, University of Maryland, College Park and BBN Technologies, July 2005.

- [Sto02] A. Stolcke, “Srilm - an extensible language modeling toolkit”, *Proc. of the 7th Int. Conf. on Spoken Language Processing, ICSLP’02*, pages. 901–904, September 2002.
- [Tak02] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, “Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world”, *3rd Int. Conf. on Language Resources and Evaluation, LREC’02*, pages. 147–152, May 2002.
- [Til00] C. Tillmann, and H. Ney, “Word re-ordering and dp-based search in statistical machine translation”, *Proc. of the 18th Int. Conf. on Computational Linguistics, COLING’00*, pages. 850–856, July 2000.
- [Til04] C. Tillman, “A unigram orientation model for statistical machine translation”, *HLT-NAACL 2004: Short Papers*, pages. 101–104, Boston, Massachusetts, USA, May 2004.
- [Til05] Christoph Tillmann, and Tong Zhang, “A localized prediction model for statistical machine translation”, *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages. 557–564, Association for Computational Linguistics, Ann Arbor, Michigan, June 2005.
- [Tur03] J.P. Turian, L. Shen, and D. Melamed, “Evaluation of machine translation and its evaluation”, *Proc. of the MT Summit IX*, September 2003.
- [Vog96] S. Vogel, H. Ney, and C. Tillmann, “Hmm-based word alignment in statistical translation”, *Proc. of the 16th Int. Conf. on Computational Linguistics, COLING’96*, pages. 836–841, August 1996.
- [Vog03] S. Vogel, Y. Zhang, F. Huang, A. Tribble, A. Venogupal, B. Zhao, and A. Waibel, “The cmu statistical translation system”, *Proc. of the MT Summit IX*, September 2003.
- [Wan98] Y. Wang, and A. Waibel, “Fast decoding for statistical machine translation.”, *Proc. of the 5th Int. Conf. on Spoken Language Processing, ICSLP’98*, December 1998.
- [Wan07] Chao Wang, Michael Collins, and Philipp Koehn, “Chinese syntactic reordering for statistical machine translation”, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages. 737–745, 2007.
- [Wat06] T. Watanabe, H. Tsukada, and H. Isozaki, “Left-to-right target generation for hierarchical phrase-based translation”, *Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, July 2006.
- [Wea55] W. Weaver, “Translation”, W.N. Locke, A.D. Booth (eds.), *Machine Translation of Languages*, pages. 15–23, MIT Press, Cambridge, MA, 1955.
- [Wu96] D. Wu, “A polynomial-time algorithm for statistical machine translation”, *34th Annual Meeting of the Association for Computational Linguistics*, pages. 152–158, June 1996.
- [Wu97] D. Wu, “Stochastic inversion transduction grammars and bilingual parsing of parallel corpora”, *Computational Linguistics*, Vol. 23, n^o 3, pages. 377–403, September 1997.

- [Xia04] F. Xia, and M. McCord, “Improving a statistical mt system with automatically learned rewrite patterns”, *Proc. of the 20th Int. Conf. on Computational Linguistics, COLING’04*, pags. 508–514, August 22-29 2004.
- [Yam01] K. Yamada, and K. Knight, “A syntax-based statistical translation model”, *39th Annual Meeting of the Association for Computational Linguistics*, pags. 523–530, July 2001.
- [Yam02] K. Yamada, and K. Knight, “A decoder for syntax-based statistical mt”, *40th Annual Meeting of the Association for Computational Linguistics*, pags. 303–310, July 2002.
- [Zen02] R. Zens, F.J. Och, and H. Ney, “Phrase-based statistical machine translation”, M. Jarke, J. Koehler, G. Lakemeyer (eds.), *KI - 2002: Advances in artificial intelligence*, Vol. LNAI 2479, pags. 18–32, Springer Verlag, September 2002.
- [Zen04] R. Zens, F.J. Och, and H. Ney, “Improvements in phrase-based statistical machine translation”, *Proc. of the Human Language Technology Conference, HLT-NAACL’2004*, pags. 257–264, May 2004.
- [Zen06] Richard Zens, and Hermann Ney, “Discriminative reordering models for statistical machine translation”, *Proceedings on the Workshop on Statistical Machine Translation*, pags. 55–63, Association for Computational Linguistics, New York City, June 2006.
- [Zha03] H. Zhang, H. Yu, D. Xiong, and Q. Liu, “Hmm-based chinese lexical analyzer ictclas”, *Proc. of the 2nd SIGHAN Workshop on Chinese language processing*, pags. 184–187, Sapporo, Japan, 2003.
- [Zha07] R. Zhang, Y. Zens, and H. Ney, “Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation”, *In Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Syntax and Structure in Statistical Translation (SSST)*, pags. 1–8, April 2007.