A proposal for an Arabic-to-English SMT system

by

Cristina España i Bonet

Advisor

Dr. Lluís Màrquez Villodre

Master in Artificial Intelligence
Universitat de Barcelona, Universitat Politècnica de Catalunya
and Universitat Rovira i Virgili

Barcelona, 15th February 2008

Contents

Chapter 1

Introduction

Language is one of the richest ways to communicate. When speaking, the information encoded in words is complemented by the intonation used, the corporal language, and possibly by external signs. Our own knowledge of the world is sometimes necessary to understand the meaning as well. In written texts, all the information must be encoded by the words, but still the meaning can be ambiguous and the environment and the culture of the reader condition the understanding of the message.

All these facts, tone, intention or knowledge, are inherent to humans but are difficult to be taken into account by machines. Natural Language Processing (NLP) is the field within Artificial Intelligence that deals with this problem. Understanding, generation or translation are general tasks treated by NLP and among them this work is focused on Machine Translation (MT).

The beginnings of MT [20, 39] date back to even before the general availability of computers, but it was not until 1949 when the Warren Weaver's memorandum [43] emphasised the possibility of using the recently invented digital computers to translate documents between a pair of natural languages. The 1950s where very optimistic and productive, but in the 1960s there was an increasing acknowledgement of the linguistic difficulties. This discouragement culminated with the report of the Automatic Language Advisory Committee (ALPAC) in 1966 [1], where it was concluded that "there is no immediate or predictable prospect of useful Machine Translation". The lack of interest and fundings due to the report in some countries such as the United States lasted almost two decades, but during the 1980s the field emerged again, now with the improvement given by faster computers and new developed NLP tools. In the 1990s, statistical approaches emerged and, nowadays, this is one of the most successful paradigm.

The improvements coincided as well with more realistic and delimited expectations of what MT can do. An MT system can be good for a given domain for instance, just as a human translator can do better in specific domains. Or in open domains, MT can help in human translations or in getting an approximate translated version of virtually any web page.

1.1 Classification of MT systems

MT systems can be classified according to their usage. Some systems are designed for machine-aided translation, both for *Human Translation with Machine Support* and for *Machine Translation with Human Support*. Here we are more interested in the third type, *Fully Automated Translation*, systems that traditionally preferred speed over quality and that were useful to get an overall idea of text contents, but that are every time more concerned about quality.

The amount and the linguistic techniques classify MT systems in *direct*, *transfer* and *interlingua* approaches. The *direct approach* does a straightforward translation word-by-word or nowadays phrase-by-phrase. The very firsts systems such as the Mark II system and the Georgetown GAT system used the direct approach for the Russian-English language pair. Later, at late 1970s, there were systems which gave more importance to linguistics. In those *transfer systems* there is a syntactic analysis of the sentences of the source language which results in an abstract representation of the sentences. This abstract representation is transferred to the abstract representation of the target language, and then, the output is generated from this representation. For the *interlingua approach* the abstract representation is assumed to be unique for every language.

Another dimension for MT classification is the architecture of the system. According to that we distinguish between *rule-based systems* and *empirical systems*. *Rule-based systems* need a group of human experts to establish the set of rules that drives the translation process. This is usually slow, expensive and not portable, but one obtains high quality syntactics for the translated output. Although both architectures deal with the three degrees of linguistic processing, rule-based systems are usually characterised by doing a syntactic or semantic analysis, and therefore perform a transfer step. On the other hand, *empirical systems* were based in their beginnings in a direct translation but some current systems perform some linguistic analysis as well. Empirical or data oriented systems need a parallel translated corpus to learn automatically and so there is no need for human contributions at least during the translation process. The learning during the training step can be of different kinds; one can learn syntactic rules or lexical translation of phrases for instance.

Within empirical systems two main approaches can be pointed out: Example-based Machine Translation (EBMT) and Statistical Machine Translation (SMT). In the first case, EBMT, new translations are formed on the basis of the previously compiled translations. In the second case, SMT, one considers that each sentence of the target language is a possible translation of a sentence in the source language and assigns a probability to each of them. The main system used in this work belongs to SMT, and Section 2.1 is a review of SMT fundamentals.

1.2 Difficulties and virtues of MT

As we have seen, natural languages are very complex by themselves. There are multiple ways of saying the same thing and a same sentence can have different meanings according to the context or the intention. Besides, every language is representative of a culture, and therefore, there are distinctive features and nuances which cannot be translated from one to another.

Synonymy and especially ambiguity are the major problems for a machine to translate. Ellipsis also make more difficult the task, since the missing information that for a human is understood, is lacking for a computer. These smaller tasks are studied independently within NLP, and their incorporation should contribute to the quality of the translation. Nowadays, however, there are not MT systems that generate general high quality translations under the point of view of a human, although good results for restricted domains can be obtained.

These facts should not discredit fully automatic MT, but one must be aware of its limitations. Translating is a very ambitious goal and computers can help in the task. An MT-translated text can help a human to understand the topic of the text in a very fast way. Or MT systems can be designed to work with a specific task and get an acceptable output. But these outputs must be revised by a human translator, as human translations are. The usefulness of MT then, lies in the relation between quality, speed and necessity. MT is useful when one needs a coarse or fast translation and when helps humans to obtain high quality results after a post-edition. This does not mean that one has to forget about getting as close as possible to a really fully automatic high quality translation.

As for the concrete case of SMT, the main advantage is that the systems are in general language independent. There is no need to develop specific tools for every language, and the *only* essential element is a parallel corpus. Of course, systems can be refined according to the language, but the core of the system remains the same. On the other hand, since they train over a corpus, the quality of the translation does depend on whether the sentences belong to the same domain of the corpus or not. That is general to all statistical-based NLP. Therefore, SMT systems are fully automatic, general, fast, and give competitive results, but the latter can be compromised when used in a different domain than the training corpus.

1.3 This work

The aim of this work is to apply MT techniques to translate from Arabic to English in the context of the 2008 NIST Machine Translation Open evaluation[1].

For the core of the system we choose a SMT architecture. With a standard SMT system we check the improvements given by adding linguistic information, that is, maximise the probability not only of the sequence of words, but of its lemma, part-of-speech and chunk as well. We increase the amount of linguistic knowledge but we also increase the sparsity in the corpus because the combination of features increases the vocabulary. We explore several approaches to these combinations.

As a second method, we use machine learning (ML) techniques to select the most adequate translation phrases and combine them with the output of the SMT system. We treat the translation task as a classification problem and use the linguistic information and the context of each word as features to train the classifiers. This

[1] `http://www.nist.gov/speech/tests/mt/2008/`

methodology is used in Word Sense Disambiguation and should help to select the correct translation of a phrase according to its context. We analyse the results of this subtask and quantify the impact in the results. The output of this phase is inserted into the SMT system by enlarging the translation table with every sense of a phrase and with the inclusion of a new probability score, which accounts for the result of the classifier. We compare the results with and without this additional information. This combination of SMT and ML, MLT, is our final proposal for the Arabic-to-English SMT system.

The outline of the report is as follows. The following chapter is devoted to summarise the basics of statistical machine translation and to sketch the main aspects of Arabic which are important in the translation process. Chapter 3 describes the data at our disposal, the preprocess applied to the corpora and the architecture of our systems. Chapter 4 shows the results for both the base SMT systems and the hybrid ones with the inclusion of the disambiguated phrases. Finally, we draw our conclusions in the last chapter and indicate some possible improvements for our final system.

Chapter 2

Background

When dealing with Machine Translation there are two main aspects that must be taken into account: the approach to be used for translating and the characteristics of the language pair involved. This chapter shows the basics of the MT models used in this work, and gives a general description of our source language, Arabic, in comparison to the target one, English.

2.1 Statistical Machine Translation

We describe in this section SMT as a fully automatic, direct, empirical system. There exists several modifications and extensions to this basic system, but next we start by summarising the foundations of a general SMT system.

2.1.1 Word-based SMT

The probabilistic approach to machine translation assigns as translation of an input sentence in a source language the sentence in the target language that maximizes probability. That is, one must generate all the possible translations for a given sentence, calculate their probability $P(output|input)$, and then explore the space to look for the most probable one. In the following, and in order to maintain the usual notation, we will denote with the letter e (from English) the translated output sentence and with f (from $f$oreing) the input one.

Even though the final goal is to find the sentence e which maximises $P(e|f)$, the current statistical machine translation models are based on the contrary assumption. The justification lies on the usage of the Bayes theorem which relates both conditional probabilities:

$$P(e|f) = \frac{P(e)\,P(f|e)}{P(f)}\,. \tag{2.1}$$

In words, the probability that e is a translation of f can be written as the product of the conditional probability of f given the input sentence e, $P(f|e)$, and the *a priori* probability of e by itself, $P(e)$. Since $P(f)$ is independent of e, it acts as just a normalization factor. Finding the sentence e that maximises $P(e|f)$ is then equivalent to maximise the likelihood:

$$T(f) = \hat{e} = \mathrm{argmax_e}\ P(e|f) = \mathrm{argmax_e}\ P(e)\,P(f|e)\,. \tag{2.2}$$

The calculation of $P(f|e)$ is in principle as hard as the calculation of $P(e|f)$, so, under this point of view, this longer path would not be worthy. The improvement comes from the fact that we are now taking into account the language model $P(e)$, that is, a collection of probability scores of word sequences in a language that takes care of the correctness and fluency of the output sentence with independence that it is a good translation or not. That for instance penalises an output sentence which is a translation of the input word by word but with an incorrect grammatical order.

These probabilities, $P(e)$ and $P(f|e)$, represent the *language model* and the *translation model* respectively. The first one only needs data in the target language to be estimated, the second one needs to extract relations from data in both the target and the source language. The third task, finding *argmax*, requires a search in a huge space which is exponential on the size of the input, and it is done by the *decoder*.

2.1.1.1 Language model

The language model is then the part that takes care of the fluency in the target language. It could be easily calculated by a count on the corpus, being the probability of a sentence e the number of times it appears in the corpus N with respect to the total number of sentences, that is, the maximum likelihood estimate:

$$P(e) = \frac{N_e}{N_{sentences}}. \tag{2.3}$$

Even though these models are not practical because the corpus is not going to have all the possible sentences, some realistic models are based on this simple idea. So as to avoid correct sentences with a null probability just because they do not appear in the corpus, the counting is not done over the whole sentence, but over small groups of n words, *n-grams*. The probability for each of these n-grams is the number of times that these words are seen together divided by the number of times the last $n-1$ words appear together. The total probability of a sentence e is the product of the probabilities of its n-grams.

The chances of finding all the n-grams within the corpus is higher that finding the whole sentence, but of course it is not assured. Since $P(e|f) \propto P(e)$ the lack of an n-gram in the corpus invalidates the sentence as an acceptable translation. This is solved by *smoothing techniques* which keep part of the probability mass to unseen n-grams.

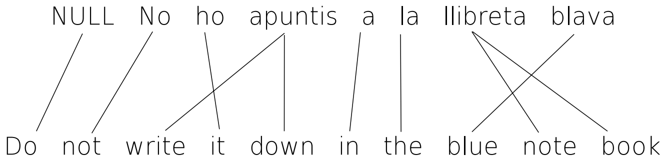
We show as an example a trigram model. When smoothing the probability of the trigram w_1, w_2, w_3 is not only $N_{w_1,w_2,w_3}/N_{w_1,w_2}$ but part of the probability mass is given to the lower order n-grams (Eq. 2.4). The dominant term (with weight λ_3) is still $N_{w_1,w_2,w_3}/N_{w_1,w_2}$, but the bigram w_2, w_3 and the word w_3 do contribute as well. In general, every word has a non-null probability even if it is not in the corpus and therefore, $P(e)$ is not going to be zero because of an unseen word in the training corpus. The weights λ_0, λ_1, λ_2 and λ_3, the smoothing coefficients, are fit with the development set.

$$P(w_3|w_1, w_2) = \lambda_3 \frac{N_{w_1,w_2,w_3}}{N_{w_1,w_2}} + \lambda_2 \frac{N_{w_2,w_3}}{N_{w_2}} + \lambda_1 \frac{N_{w_3}}{N_w} + \lambda_0 \tag{2.4}$$

There are several smoothing techniques. The one which just served as example is a lineal interpolation where for every n-gram the lower order ones are also used. Others such as the *back-off* models, only use the lower orders when the n-gram is not in the corpus. The calculations in this work use an interpolation that discounts an amount from each n-gram in the corpus. The probability of the low order terms is in this case proportional to the number of the different words that precede it (*Kneser-Ney* smoothing [22, 11]). That makes loose weight to words such as *York* if they always appear preceded by the same word, as it happens with *New*.

2.1.1.2 Translation models: IBM models

The second component involved in the process of translation is the translation model $P(f|e)$ (see Ref. [5] for a review). To estimate it, it is necessary to extract the information from an aligned parallel corpus with a one to one correspondence between the source and target languages. In an intuitive way, one can see the important ingredients to model the translation:



One needs to know the translation of every word and the number of words necessary in the target language, the position they occupy within the sentence and the number of words that need to be generated. So, the translation model has several contributions:

- the probability that *blava* generates the translation *blue* $t(blue|blava)$: *lexical probability*,
- the probability that *blava* generates x words $n(x|blue)$: *fertility*,
- the probability that *blava* in the i position generates *blue* in the j position $d(j|i,m,n)$: *distortion* (where m is the length of the input sentence and n of the output one),

- and the probability that a spurious word is generated p_1. These words are generated from the NULL position which is assigned as the zeroth position of every input sentence.

Alignments

All of these contributions could be calculated by a straightforward counting with parallel corpora aligned not only at a sentence level but at a word level too. Since this kind of corpus is not available, one must construct the alignments as a first step of every translation model.

An alignment is represented by a vector of integers, with length the number of translated words and with every component indicating the position of the word in the original sentence.

The probability of one alignment a, $P(a, f|e)$, is a function of the words in the sentence pair and the probability tables n, p, t, and d as introduced at the beginning of the section [5]. The final translation probability of the sentence pair f-e is the sum of that of every possible alignment and is estimated via unsupervised learning from a corpus:

$$P(f|e) = \sum_a P(a, f|e) \tag{2.5}$$

Models

The firsts models for SMT defining $P(f|e)$ were proposed at the beginning of the 90's by Brown et al. [5]. These so-called IBM models are still widely used. The models go from 1 to 5 in an increasing complexity. Given the prohibitive number of parameters to be determined during translation, the first models, with strong enough assumptions so as to allow exact calculations, are used as the seed for the last and more reliable models.

Model 1 uses the translation probability alone $t(f_i|e_i)$; the length of the translated sentence is fixed with all the possible lengths equiprobable, the fertility is assumed to be 1 for every word, NULL included, and all distortions with the length of the sentence known are equiprobable. With these simplifications, the iterative Expectation-Maximization algorithm [12] drives to an absolute and unique minimum which does not lead to good alignment probabilities.

Model 2 introduces a slight improvement: it takes into account distortion and, therefore, the position of the word within the sentence. Model 1 is just a particular case where distortion is fixed to $1/(n+1)$[1]. This way, Model 1's results can be taken as initial parameters for Model 2.

Model 3 uses a combination of translation probabilities, distortion and fertilities. However, since the distortion probabilities are independent from one word to another, some positions can be occupied by several words and others remain empty. The initial values for $t(f_i|e_i)$ and $d(j|i,m,n)$ can be those given by Model 2.

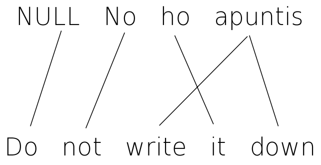
Model 4 is already a step towards a translation based on *phrases* instead of words. We define as a phrase a group of words that usually go together and must, therefore, move together within the sentence. This changes the form of the distortion probability and two components are needed: one indicating the position of the *head* words and another one for the others.

Finally, Model 5 takes into account that two translated words cannot occupy the same position.

2.1.2 Phrased-based SMT

The model introduced in the previous section is the core of the current state-of-the-art of Statistical Machine Translation. However, it was soon noticed that translation is not a word to word process, and that the information of surrounding words would help and that one word could be translated into more than one element. This motivated the usage of *phrases* as translation units. Within this context, a phrase is a sequence of words that appear together in the source sentence, but it is not necessarily defined according to the syntactic structure of the sentence (see an example in Figure 2.1).

[1] Distortion in Model 2 and the one defined in the previous section are defined as inverse conditional probabilities.



No ⟷ Do not
No ⟷ not
ho ⟷ it
No ho apuntis ⟷ Do not write it down
ho apuntis ⟷ write it down

Figure 2.1: Example of all the extracted phrases which are coherent with the shown alignments.

The firsts attempts to consider *phrases* as the atomic units defined the phrases from word alignments, that is, two phrases are aligned when its words are only aligned within its limits and never outside the phrase [37]. Once the alignment between phrases is established, the word alignments are not necessary any more [25].

Word alignment is an active field of research. There exist several heuristics used to combine the alignments obtained from the two translation directions to improve the final result:

- *Intersection* of the alignments in the two directions. It is the most restrictive combination, and, therefore, gives high-precision alignments and the largest number of extracted phrases.
- Contrary to the previous one, the *union* produces less phrases due to the larger number of alignment points.
- Starting with the intersection and adding some additional alignment points, some *grow* heuristics refine the final alignment. The heuristics *grow* [36] and *grow-diag-final* [25] are used in this work and add those points belonging to the union that connect at least one word which was not yet aligned.

Once selected all the phrases consistent with the alignments, the phrase translation probability can be calculated by relative frequency. Each source phrase $\bar{f}_i$ is then translated into one phrase in the target language $\bar{e}_i$, and afterwards the output phrases are reordered according to the distortion probability.

There are several extensions to phrase-based models. In the following, we introduce two of them, the ones being used in this work.

2.1.2.1 Log-linear Model

The Log-linear Model is an extension to the original phrase-based approach. It uses the fact that the maximum likelihood estimate equals the maximum entropy one in order to move from the product of probabilities, Eq. 2.2, to a linear sum of its logarithms h_m [34]:

$$\text{argmax } P(e|f) = \text{argmax } \sum_m \lambda_m h_m(f|e)\,. \tag{2.6}$$

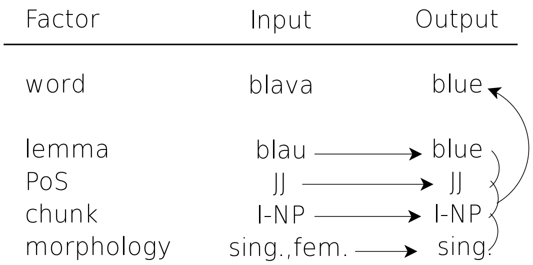
Rewritten in this way, it is easy to include additional information. Besides the language and translation model, this extra information can be a distortion model $P_{di}(e,f)$ accounting for the amount of reordering, or a word (phrase) penalty model, accounting for the length of the output. Word penalty $w(e)$ takes care of the length of the whole translated sentence and phrase penalty $ph(e)$ of the average length of phrases [44]. In both cases negative values favour longer outputs. All these additional models are described by a function $h(\cdot)$ and its corresponding weight λ is adjust with an independent development set with the minimum error rate training for instance.

2.1.2.2 Factored Translation Models

Another approach designed to include additional information are the so-called *factored models* [23]. This extension to phrase-based models considers instead of a single word a vector of factors each of them representing a feature with some linguistic information. This extra information might be morphological, syntactic or semantic; it can include the lemma, part of speech, chunk label, statistically derived word classes, case, gender or whatever feature relevant to the language pair to be translated. This can be useful for morphologically rich languages such as Arabic;

however, the larger the vector of factors the slower the translation process, and even more important, it sometimes represents a prohibitive time for training.

The process of translation is divided here in several mapping steps of two kinds. The first kind translates source factors into the target ones. The second kind generates the final surface form of the word in the target language given the set of linguistic factors already in the target language. For example, one could first translate lemmas and morphological information separately and then generate the surface word given that translated lemma and morphology:



All the components are combined in a log-linear model. Every translation and generation step is treated as a function $h(f|e)$ as the language model or the reordering model are. This kind of models, factored models and in general log-linear models, have been implemented in the `Moses` package [26].

2.2 Hybrid Machine Translation

We have seen in the Introduction that SMT systems are not the only approach to machine translation, and that the branch of transfer systems is as important and effective at least in some language pairs. This is especially valid for languages with very different syntax and rich morphology. It can be therefore interesting to get the best aspect of each approach to improve the final result. Given the large amount of models which combine both approaches or others available, we focus here in those that make use of Word Sense Disambiguation techniques (WSD) to help in the translation process within a SMT system.

2.2.1 Discriminative phrase selection

The hybrid systems used in this work are based on a SMT one, but we use linguistic information and the surrounding context in order to translate phrases. It is not a Syntax-based System, but it uses WSD techniques to select the phrases. The general WSD task tries to identify which sense of a word must be used in a given sentence. Here, we understand the different translations of a phrase as different senses of that phrase, and try to identify which one is the most adequate given the sentence. Contrary to factored models, this allows to take into account the context of each phrase to translate it, and phrase selection is treated as a classification problem instead of a translation probability given by relative frequency counts.

There are several recent methods in the literature to integrate WSD techniques into the translation process. In 2005, Carpuat and Wu [8] used the WSD predictions to constrain the possible translations available in decoding time. In the same year Vickrey et al. [42] applied discriminative models for word selection but used in a blank-filling task instead of full translation. This work was first extended to the full translation task [6, 7] and afterwards to translate phrases instead of words [9, 10].

Carpuat and Wu, the authors of Ref. [9], have developed a WSD system which combines naïve Bayes, maximum entropy, boosting and kernel PCA-based models. Bangalore et al. [3] rely on a maximum entropy model. Here, we use the model of Giménez and Màrquez [19] based on the use of Support Vector Machines (SVM) to solve the multi-class classification problem where every possible translation is a class.

In that model, given the phrases extracted from the parallel corpus, each occurrence of a phrase is taken as a positive example for its current translation and negative for the rest. This way the multi-class problem is binarized and converted in a one-vs-all decision as it is graphically seen in Figure 2.2. The feature set for each example contains information of the source phrase such as lemma, PoS, and chunk labels for the phrase itself and a context, let's say 5 words to the left and to the right, by taking n-grams of the linguistic information.

The result of this model as to its application to machine translation is a probability table $P_{\text{DPT}}(e|f)$. However, not every phrase will have a DPT (Discriminative Phrase Translation) prediction, since the number of examples must be reasonable

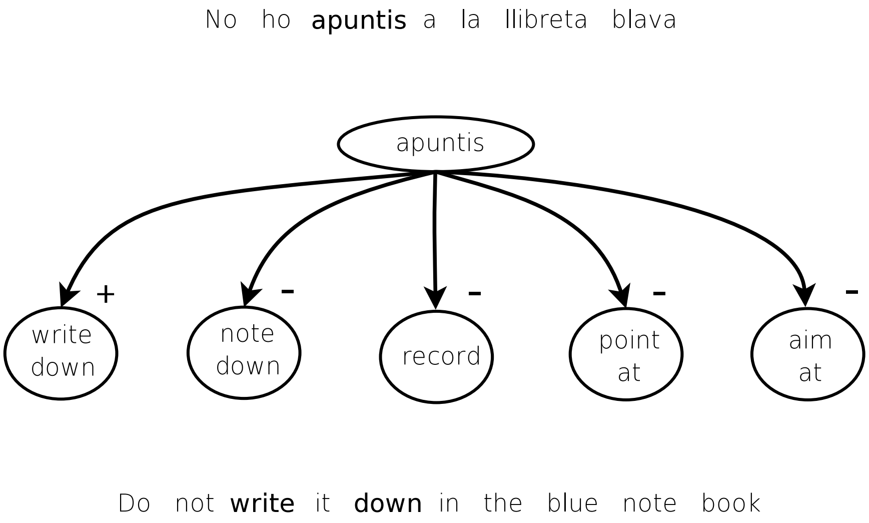

No ho apuntis a la llibreta blava
apuntis
+
-
-
-
-
write down
note down
record
point at
aim at
Do not write it down in the blue note book

Figure 2.2: Example of the translation of the phrase *apuntis*. The true translation is a positive example to train the SMV; the other possible translations are negative examples.

to train the classifier and the table must be completed with the standard MLE (Maximum Likelihood Estimation) table. The final probability is included in the translation system as a component of a log-linear model:

$$\begin{aligned}\log P(e|f) \;\propto\;& \lambda_{lm}\log P(e) + \lambda_{lg}\log lex(f|e) + \lambda_{ld}\log lex(e|f)\\ &+\lambda_{g}\log P_{\text{MLE}}(f|e) + \lambda_{d}\log P_{\text{MLE}}(e|f) + \lambda_{\text{DPT}}\log P_{\text{DPT}}(e|f)\\ &+\lambda_{di}\log P_{di}(e,f) + \lambda_{ph}\log ph(e) + \lambda_{w}\log w(e)\,, \qquad (2.7)\end{aligned}$$

where $P(e)$ is the language model probability, $lex(f|e)$ and $lex(e|f)$ are the generative and discriminative lexical translation probabilities respectively, $P_{\text{MLE}}(f|e)$ the MLE generative translation model, $P_{\text{MLE}}(e|f)$ the discriminative one, $P_{\text{DPT}}(e|f)$ the DPT model, $P_{di}(e,f)$ the distortion model and $ph(e)$ and $w(e)$ correspond to the phrase and word penalty models.

2.3 Evaluation

Evaluation in MT is an active research field since it is difficult to define what makes good a translation given that there is not a unique translation for an input. A system can be evaluated both by humans or automatically.

Manual evaluation is slow and subjective, but one can qualify aspects which cannot be evaluated by a computer. On the other hand, *automatic* evaluation uses objective metrics that allow for a fast qualification of the translation, but not every detail can be grasped by a metric. In order to compare in an objective and fast way several systems, this work uses an automatic evaluation, but one must take into account that that is just comparing the aspects that the used metric does.

There exists several metrics. We focus here in those based on the lexical similarities (number of coincident n-grams) between the automatic translation and human reference translations. Just to name some WER [31], PER [41], NIST [14], GTM [30], ROUGE [28], BLANC [29], METEOR [2] and BLEU [38] are n-gram based metrics. BLEU (Bilingual Evaluation Understudy) is one of the most used metrics and the one we use as a reference in our results. It calculates an score that depends on the coincident n-grams up to order 4. As all these metrics have in common, this score only takes into account how fluent the output is and how equal to a reference is, but it does not evaluate if the translation captures the meaning of the input.

2.4 Language pair

يولد جميع الناس أحراراً متساوين في الكرامة والحقوق. وقد وهبوا
عقلاً وضميراً وعليهم ان يعامل بعضهم بعضاً بروح الإخاء.

All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

(Article 1 of the Universal Declaration of Human Rights)

The nature of the language pair is an important aspect in the translation process, and one should refine and adapt the general machine translation system in order to catch its peculiarities. In the following, we sketch the main features of the two languages involved in our translation system. Due to the larger differences with Latin languages, this section focuses on Arabic with just some insights into English.

2.4.1 Arabic

Arabic is a Semitic language belonging to the Afro-Asiatic family. More that 200 million people speak one of the numerous Arabic dialects, and all of them have as a standard language the Koranic one. After some modifications the language in the Koran has evolved from being the classical Arabic to the be considered the modern literary language.

As all of the Semitic languages, Arabic is written from right to left and from top to bottom. Numbers, however, are written from left to right in the right to left text. In Modern Standard Arabic, numbers are usually written as *Indian numerals* while it is in Moroccan Arabic that numbers are written as what we call *Arabic numerals*. Corpora for MT mix both forms:

٠	١	٢	٣	٤	٥	٦	٧	٨	٩	١٠
0	1	2	3	4	5	6	7	8	9	10

The syntax, contrary to Catalan or English for instance, follows a VSO structure (verb-subject-objects). There are also copular sentences without any verb.

The Arabic script is an alphabet with allographic variants, diacritics and ligatures. Each character has four allographs depending on its position within the word: initial, medial, final or as stand alone. The alphabet is composed by 25 consonants, 3 semi-consonants, 3 short vowels, 3 long vowels and 2 diphthongs. The short vowels, *fatha*, *kasra* and *damma*, are not letters themselves but diacritics written above or below consonants. Other diacritics are also used as a non-vowel mark (*sukun*), as a double consonant mark (*shadda*), or as a letter itself (*hamza*). Figure 2.3 shows some examples of the diacritics when added to the letter *baa*.

However, diacritics are not usually seen in written texts. They appear in the

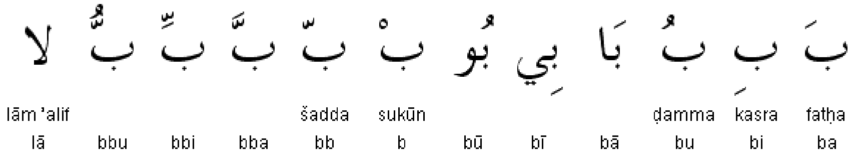


Figure 2.3: Diacritics used in the Arabic script, here added to the letter *baa*.

Koran, in some other religious texts, classical poetry, textbooks or in complex texts to avoid ambiguity. However, in most cases, when pronunciation is not especially important, texts are non-vocalized and non-diacritized. This is mostly the case of the corpora used for MT. Another character to comment is *tatweel*, used as elongation for text highlight or justification. It can be therefore eliminated from the corpora before training.

Appendix A lists the characters and shows the Arabic glyphs. The same table shows the Buckwalter transliteration which will be introduced in Section 3.3.1 as the commonly used romanization system in NLP. Romanization is useful to equate Arabic and Latin scripts in order to be treated homogeneously by machines. Besides, it eases the understanding for those not familiarised with the Arabic phonetics.

Words are formed by combination of the previous elements sometimes joined together by ligatures. A full word agglutinates to the root affixes and clitics. Affixes mark tense, genus and number. Clitics are divided in proclitics (before the root) and enclitics (at the end of the word). Proclitics are prepositions, conjunctions and determiners; enclitics are pronouns and possessives.

Let us see an example. The syntactic phrase *and by their virtues* is written in Arabic as an only word وبحسناتهم (or *wbHsnAthm* using Buckwalter's transliteration). The word can be morphologically segmented as:

enclitic	affix	stem	proclitics	
hm	At	**Hsn**	b	w
(their)	(s)	(virtue)	(by)	(and)

where it is taken into account that Arabic is read from right to left. More examples and the full set of clitics are introduced with the tokenization of the corpora in Section 3.3.1.

2.4.2 English

English is an Indo-European language with Latin writing. It is spoken by more than 300 million people as first language and it is usually the target language for translation when the system is not designed for a concrete purpose.

Besides the fact that, contrary to Arabic, English it is written from left to right, syntax has SVO structure (subject-verb-objects). That makes reordering important in the translation Arabic-to-English. At the level of words, the English grammar has minimal inflection at least compared with such a morphologically rich language as Arabic.

There are numerous syntactic details different between both languages and most of them will be catch statistically. As we have said, one of the advantages of SMT is that it is in principle a language independent system capable of capturing the peculiarities of every language.

Chapter 3

System design

This chapter describes the data at our disposal to build the Arabic-to-English translation system and the software used for the different tasks. It also reports the preprocessing applied to the raw data and the architecture of the two systems used in this work.

3.1 Parallel corpora

Parallel corpora are needed in order to estimate the translation models. In the following, we use corpora belonging to two domains: news and transcriptions from the United Nations.

3.1.1 News data compilation

The training set is a compilation of six corpora supplied by the Linguistic Data Consortium (LDC) for the 2008 NIST Machine Translation Open evaluation. The sources for these corpora are the Agence France Press News Service, An Nahar, Assabah, Xinhua News Service, Language Weaver News, and Ummah Press Service. From the whole corpus, lines[1] with a length shorter than 100 words and not

[1] Each line corresponds to the minimum aligned unit. The aligments are given at a fragment level, which is in most cases larger than one sentence.

more than nine times longer in one language than in the other one are used in the compilation. That is the optimal length for training the `Moses` decoder[2] and the length ratio limit for obtaining the alignments with `GIZA++`[3]. With this, 123,662 lines, a 99% of the total, have been obtained, resulting a medium size corpus under the point of view of collecting alignments. Table 3.1 shows the corpora with the corresponding identification, the number of lines used and the equivalence in words for the English and the Arabic parts. The concrete specifications can be read from the LDC Corpus Catalogue[4].

For the development and test sets we selected 500 lines from the same corpora with the exception of the *Multiple-Translation Arabic* and the *TIDES MT2004 Arabic evaluation data*. The former is a collection of files with 7, 12 or translations only used for training. The latter is a small compilation coming from sources included in the other corpora as well. Table 3.2 shows the details of the samples and the number of lines from each corpus which is proportional to the one in the training set.

3.1.2 United Nations corpus

Outside the news domain, the corpus of the United Nations offers a great amount of data ranging from year 1993 to 2002. We have neglected some damaged lines corresponding to year 2001 and eliminated a fragment written in Russian instead of Arabic in year 1997. After that we apply the same cleaning as for the news set, i.e. cutting the lines with more than 100 words or more than nine times longer in one language than in the other one. With this preprocess we obtain a parallel corpus with 3,686,372 lines.

We have done three different partitions on the resulting corpus. An small one with 20,000 lines used for studying the impact of translation with factored models with information of lemma, PoS and chunks. A second corpus with 125,000 lines comparable to the news corpus. Finally, we consider a large corpus with 3,400,000 lines. In the three cases we keep 500 lines for development and 500 more for testing.

[2] http://www.statmt.org/moses/

[3] http://www.fjoch.com/GIZA++.html

[4] http://www.ldc.upenn.edu/Catalog/

Corpus	LDC id	Lines	Tokens (Arabic)	Tokens (English)
Arabic English Parallel News Part 1	LDC2004T18	61,000	2,179,289	2,273,021
Arabic News Translation Text Part 1	LDC2004T17	18,000	532,771	602,262
Arabic Treebank English Translation	LDC2005E46	23,800	660,821	739,695
eTIRR Arabic English News Text	LDC2004E72	4,000	97,882	98,655
Multiple-Translation Arabic	LDC2003T18	15,533	434,465	507,617
(Part 1 & 2)	LDC2005T05			
TIDES MT2004 Arabic evaluation data	LDC2006E44	1,329	40,667	47,324
Total:		123,662	3,945,895	4,262,740

Table 3.1: Detailed composition of the training data set used for the news translation task.

Corpus	LDC id	Lines	Tokens (dev)		Tokens (test)	
			Arabic	English	Arabic	English
Arabic English Parallel News Part 1	LDC2004T18	280	11,575	11,513	10,740	10,641
Arabic News Translation Text Part 1	LDC2004T17	100	2,776	3,141	2,856	3,276
Arabic Treebank English Translation	LDC2005E46	100	2,532	2,875	1,355	1,411
eTIRR Arabic English News Text	LDC2004E72	20	625	485	523	544
Total:		500	17,508	18,014	15,474	15,872

Table 3.2: Detailed composition of the development and test data sets used for the news translation task.

3.2 Monolingual corpora

A monolingual corpus is needed in order to estimate the language model. Each of the parts of a parallel corpus can be used as a monolingual data set, and we calculate the language model from these data as explained in Section 3.4.

3.3 Linguistic processing

Before using the corpora for MT some linguistic preprocessing must be applied. We divide the process in two steps. First, the input is converted to a unique codification. The fact that Arabic and English have two different scripts makes the translation process harder, and so, we choose to convert the Arabic glyphs to the Latin alphabet as explained in the following. As a second step, we tokenize both of the input languages and annotate them with the lemma, part of speech and chunk label for each word. The concrete tools depend on the language as well.

3.3.1 Arabic

There exist several transliterations to convert Arabic characters to the Latin alphabet. In NLP, the original text encoded in ISO-8859-6 or CP-1256 for example are usually converted to the Buckwalter transliteration. That is a one to one correspondence between Unicode and UTF-8 codification. Appendix A shows this correspondence between the Arabic glyphs, the Unicode symbol and the Buckwalter UTF-8 character.

We alter the standard transliteration by using the *XML-friendly* version which changes the characters <, > and & to I, O and W respectively. That allows to generate the XML files necessary for the discriminative learning without problems. The character for *madda*, $|$, is a reserved character in the `Moses` decoder that separates the different factors for a word. Therefore, it has been substituted by L after the annotation process.

Note as well that actual presentation glyphs vary with context as well as entering into various ligatures. Some of these ligatures such as لا, لآْ, لاَ or لأ have not been detected in the automatic transliteration but converted afterwards.

The standard Buckwalter transliteration has been a prerequisite necessary to annotate the Arabic part of the corpus using the `ASVMTools` [13]. This software uses the `Yamcha SVM tools` [27] to tokenize, PoS tag and Base Phrase Chunk the input text. `ASVMTools` includes models trained on the Arabic Penn TreeBank ATB 1 v3.0, ATB 2 v2.0 and ATB 3 v2.0, therefore on a news domain. Finally, since the public version of `ASVMTools` does not separate the determiner *Al* (the), we have separated it after the annotation process. This separation has been done over all the words beginning with *Al* unless over those already appearing in the Arabic WordNet as full words, keeping the information of lemma, part of speech and chunk label.

In the following, we show the annotation process and deep into the details for a segment belonging to the Arabic English Parallel News Part 1:

$< seg \;\; id = 18 >$
وتأتي دول فرنسا وبريطانيا وايطاليا وألمانيا وأيرلندا وأسبانيا ولوكسمبورج في المقدمة
وبخلاف الشركات الأوروبية فقد وصل حجم رءوس الأموال المصدرة للشركات العاملة في
مصر حتي ديسمبر ٢٠٠٠ الي ١٢٦ مليار ج نيه لعدد ١٠ آلاف شركة استثماري ة مؤسسة
وفقا لقانون الاستثمار ..
$< /seg >$

Using the standard Buckwalter transliteration, the above text is converted into:

```
  wt>ty dwl frnsA wbryTAnyA wAyTAlyA w>lmAnyA w>yrlndA w>sbAnyA
 wlwksmbwrj fy Almqdmp wbxlAf Al$rkAt Al>wrwbyp fqd wSl Hjm r'ws
Al>mwAl AlmSdrp ll$rkAt AlEAmlp fy mSr Hty dysbmbr 2000 Aly 126 mlyAr
 jnyh lEdd 10 |lAf $rkp AstvmAryp m&ssp wfqA lqAnwn AlAstvmAr ..
```

Tokenization

The process of tokenization segments the words in proclitics, stems+affixes, and enclitics. Punctuation is considered as an independent token as well. Arabic proclitics are prepositions *b* (by/with), *l* (to) and *k* (as); conjunctions *w* (and) and *f* (then); and the determiner *Al* (the). All of them unless the determiner *Al* which is not separated in the Arabic TreeBank have been segmented by the `ASVMTools` tokenizer. The set of enclitics comprises the pronouns and possessive pronouns: *y* (my/mine), *nA* (our/ours), *k* (your/yours), *kmA* (your/yours masc. dual), *km* (your/yours masc. pl.), *knA* (your/yours fem. dual), *kn* (your/yours fem. pl.), *h* (him/his), *hA* (her/hers), *hmA* (their/theirs masc. dual), *hnA* (their/theirs fem. dual), *hm* (their/theirs masc. pl.) and *hn* (their/theirs fem. pl.).

An Arabic word may be composed by a conjunction, a preposition and the determiner at the beginning of the word, as proclitics; the stem and its affixes and one pronoun at the end, as enclitic. `ASVMTools` attack the tokenization task as a 1-of-6 classification task for each letter. As an example, our running text converts into:

```
w t>ty dwl frnsA w bryTAnyA wAyTAlyA w >lmAnyA w >yrlndA w >sbAnyA w
  lwksmbwrj fy Almqdmp w b xlAf Al$rkAt Al>wrwbyp f qd wSl Hjm r'ws
  Al>mwAl AlmSdrp l Al$rkAt AlEAmlp fy mSr Hty dysbmbr 2000 Aly 126
mlyAr jnyh l Edd 10 |lAf $rkp AstvmAryp m&ssp wfqA l qAnwn AlAstvmAr
```

We have highlighted in blue the segmented clitics. Green is used to indicate other segmentations learned from the Arabic Penn TreeBank, in this case the change from *ll* to *l Al*.

Feminine lemmatization

We do not apply a true lemmatization to the corpus. The affixes are not separated from the stem, but we only restore the feminine singular marker *p* instead of a *t* after decliticization. We consider our final tokens the result of this step.

Part of Speech tagging

Following the Arabic TreeBank distribution, the `ASVMTools` use the 24 PoS tags from the collapsed tag set. This is now a 1-of-24 classification task with class labels:

CC	Coordinating conjunction	NUMERIC_COMMA	
CD	Cardinal number	PRP	Personal pronoun
CONJ+NEG_PART		PRP$	Possessive pronoun
DT	Determiner	PUNC	Punctuation
FW	Foreign word	RB	Adverb
IN	Prep./subord. conjunction	RP	Particle
JJ	Adjective	UH	Interjection
NN	Noun, singular or mass	VBD	Verb, past tense
NNS	Noun, plural	VBN	Verb, past participle
NNP	Proper noun, singular	VBP	Verb, present
NNPS	Proper noun, plural	WP	Wh-pronoun
NO_FUNC	No function	WRB	Wh-adverb

Thus the tags account for the singular/plural distinction in nouns, but the distinction of number and gender in verbs is not reflected.

Base Phrase chunking

With the PoS tagged text, the last learning does a 1-of-19 classification task to to chunk the phrases according to IOB tagging scheme (Inside-Outside-Beginning). That applied to ADJP, ADVP, CONJP, NP, PP, PRT, SBAR, UCP and VP conforms the 19 tags.

Final annotated text

The labels of the two previous steps are added to the original word. We also replicate the word in order simulate the position of the lemma which is not obtained for Arabic but it is for English. As a result we rewrite the Arabic part of the corpus with the format *word*|*lemma*|*PoS*|*chunk* suitable for factored models in `Moses`. The separator "|" makes us substitute the character from | to *L*. Now, after the use of the trained models from `ASVMTools`, one can convert the standard Buckwalter Transliteration to the XML-friendly version as indicated in blue:

```
w|w|CC|O tOty|tOty|VBP|B-VP dwl|dwl|NN|B-NP frnsA|frnsA|NNP|B-NP
 w|w|CC|O bryTAnyA|bryTAnyA|NNP|B-NP wAyTAlyA|wAyTAlyA|JJ|I-NP
```

```
w|w|CC|O OlmAnyA|OlmAnyA|NNP|B-NP w|w|CC|O OyrlndA|OyrlndA|NNP|B-NP
w|w|CC|O OsbAnyA|OsbAnyA|NNP|B-NP w|w|CC|O
lwksmbwrj|lwksmbwrj|NNP|B-NP fy|fy|IN|B-PP Almqdmp|Almqdmp|NN|B-NP
w|w|CC|B-PP b|b|IN|B-PP xlAf|xlAf|NN|B-NP Al$rkAt|Al$rkAt|NNS|B-NP
AlOwrwbyp|AlOwrwbyp|JJ|I-NP f|f|CC|B-ADVP qd|qd|RP|B-PRT
wSl|wSl|VBD|B-VP Hjm|Hjm|NN|B-NP r’ws|r’ws|NN|B-NP
AlOmwAl|AlOmwAl|NN|B-NP AlmSdrp|AlmSdrp|JJ|B-ADJP l|l|IN|B-PP
Al$rkAt|Al$rkAt|NNS|B-NP AlEAmlp|AlEAmlp|JJ|I-NP fy|fy|IN|B-PP
mSr|mSr|NNP|B-NP Hty|Hty|IN|B-PP dysbmbr|dysbmbr|NN|B-NP
2000|2000|CD|B-NP Aly|Aly|IN|B-PP 126|126|CD|B-NP mlyAr|mlyAr|NN|I-NP
jnyh|jnyh|NN|I-NP l|l|IN|B-PP Edd|Edd|NN|B-NP 10|10|CD|B-NP
LlAf|LlAf|NN|I-NP $rkp|$rkp|NN|I-NP AstvmAryp|AstvmAryp|JJ|I-NP
mWssp|mWssp|NN|B-NP wfqA|wfqA|NN|B-NP l|l|IN|B-PP qAnwn|qAnwn|NN|B-NP
AlAstvmAr|AlAstvmAr|NN|B-NP .|.|PUNC|O .|.|PUNC|O
```

Finally, we have manually separated the determiner *Al*. We keep the annotation labels obtained from `ASVMTools` for all the words beginning with *Al-*, but in case the full word does not appear in the Arabic WordNet we segment out the determiner and adapt the chunk label as adequate. We have extracted 309 words from the Arabic WordNet beginning with *Al-*. However, since we are comparing the stem and not the lemma with those words, there is a loss in the precision of the segmentation. For illustration purposes, we have highlighted in blue the segmented words:

```
w|w|CC|O tOty|tOty|VBP|B-VP dwl|dwl|NN|B-NP frnsA|frnsA|NNP|B-NP
w|w|CC|O bryTAnyA|bryTAnyA|NNP|B-NP wAyTAlyA|wAyTAlyA|JJ|I-NP
w|w|CC|O OlmAnyA|OlmAnyA|NNP|B-NP w|w|CC|O OyrlndA|OyrlndA|NNP|B-NP
w|w|CC|O OsbAnyA|OsbAnyA|NNP|B-NP w|w|CC|O
lwksmbwrj|lwksmbwrj|NNP|B-NP fy|fy|IN|B-PP Al|Al|DT|B-NP
mqdmp|mqdmp|NN|I-NP w|w|CC|B-PP b|b|IN|B-PP xlAf|xlAf|NN|B-NP
Al|Al|DT|B-NP $rkAt|$rkAt|NNS|I-NP Al|Al|DT|I-NP
Owrwbyp|Owrwbyp|JJ|I-NP f|f|CC|B-ADVP qd|qd|RP|B-PRT wSl|wSl|VBD|B-VP
Hjm|Hjm|NN|B-NP r&apos;ws|r&apos;ws|NN|B-NP Al|Al|DT|B-NP
OmwAl|OmwAl|NN|I-NP Al|Al|DT|B-ADJP mSdrp|mSdrp|JJ|I-ADJP l|l|IN|B-PP
Al|Al|DT|B-NP $rkAt|$rkAt|NNS|I-NP Al|Al|DT|I-NP EAmlp|EAmlp|JJ|I-NP
fy|fy|IN|B-PP mSr|mSr|NNP|B-NP Hty|Hty|IN|B-PP
dysbmbr|dysbmbr|NN|B-NP 2000|2000|CD|B-NP Al|Al|DT|B-PP y|y|IN|I-PP
```

```
126|126|CD|B-NP mlyAr|mlyAr|NN|I-NP jnyh|jnyh|NN|I-NP l|l|IN|B-PP
Edd|Edd|NN|B-NP 10|10|CD|B-NP LlAf|LlAf|NN|I-NP $rkp|$rkp|NN|I-NP
AstvmAryp|AstvmAryp|JJ|I-NP mWssp|mWssp|NN|B-NP wfqA|wfqA|NN|B-NP
l|l|IN|B-PP qAnwn|qAnwn|NN|B-NP Al|Al|DT|B-NP AstvmAr|AstvmAr|NN|I-NP
.|.|PUNC|O .|.|PUNC|O
```

Notice that the separation of determiners increases the length of the sentence. Before any processing, the original sentence has 42 tokens. The number grows up to 54 when the clitics are segmented out, and up to 64 when also are the determiners. This is just a representation of the global behaviour. The mean length of a sentence in the news corpus is initially of 27.4 words increasing to 31.8 in the first case and to 38.2 in the second. That has consequences when cleaning the corpus because the length of the English sentence remains the same, a mean of 34.5 tokens per sentence. The limit of `GIZA++` for the ratio between the lengths of the sentences for calculating the alignments eliminates more sentences the more we segment the original text. In this second case where both clitics and determiners have been separated from the stem, we have kept sentences shorter than 120 words instead of the 100 words limit of the other cases. With this we obtain three corpora in the news domain differentiated by the level of segmentation:

	lines	tokens	toks/line
punct.	124,154	3,402,824	27.4
punct.+clitics	123,662	3,939,726	31.8
punct.+clitics+Al	123,498	4,718,933	38.2

3.3.2 English

The preprocessing with the English language is simpler than for Arabic since it is its codification the one used as a reference. Then, the only preprocess before annotating the corpus has been to lowercase and tokenize the sentences.

As before, the linguistic information is added to the probabilistic translation by considering the lemma, part of speech and chunk position of every word. First, lemma and PoS have been obtained with SVMTool [18], and Yamcha [27] is used afterwards for BP chunking. These tools have been trained with the Wall Street Journal (WSJ) corpus.

The corresponding translation to the example sentence in Arabic is a line composed by three sentences:

```
France, Britain, Italy, Germany, Ireland, Spain, and Luxembourg came
first.  A part from the European companies, the issued capital of
companies operating in Egypt reached LE126 billion up till December
2000.  Such capital is of 10,000 investment companies set up under
the investment law.
```

Tokenization

Since in English there is no difference between lowercase and uppercase letters as there is in the Buckwalter transliteration, all the English corpus has been lowercased. The text has been tokenized using the perl script of Josh Schröder provided by the ACL 2007 Second Workshop on Statistical Machine Translation[5]. This script separates punctuation keeping it together in numbers and some abbreviations.

```
france , britain , italy , germany , ireland , spain , and luxembourg
came first.  a part from the european companies , the issued capital
of companies operating in egypt reached le126 billion up till
december 2000.  such capital is of 10,000 investment companies set up
under the investment law .
```

Lemmatization

On the contrary to Arabic, the English corpora have been lemmatized. We use a table with with 185,201 entries where each word is listed with its lemma according to its part-of-speech.

```
france|france ,|, britain|britain ,|, italy|italy ,|, germany|germany
,|, ireland|ireland ,|, spain|spain ,|, and|and luxembourg|luxembourg
came|come first.|first.  a|a part|part from|from the|the
```

[5] http://www.statmt.org/wmt07/

```
european|european companies|company ,|, the|the issued|issue
capital|capital of|of companies|company operating|operate in|in
egypt|egypt reached|reach le126|le126 billion|billion up|up till|till
december|december 2000.|2000.  such|such capital|capital is|be of|of
10,000|10,000 investment|investment companies|company set|set up|up
under|under the|the investment|investment law|law .|.
```

Part of Speech tagging

Although being a less rich language than Arabic, the tagset labels for English is larger than the collapsed tagset we use for Arabic. This is because using the full Buckwalter's tagset for Arabic increases too much the sparsity and better results were obtained with a collapsed set obtained from a mapping from the Arabic POS tagset to Penn English. However, for English we can use de full tagset from the Wall Street Journal with 36 labels:

CC	Coordinating conjunction	PP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Prep./subord. conjunction	SYM	Symbol (mathematical or scientific)
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund/present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd ps. sing. present
NNP	Proper noun, singular	VBZ	Verb, 3rd ps. sing. present
NNPS	Proper noun, plural	WDT	wh-determiner
PDT	Predeterminer	WP	wh-pronoun
POS	Possessive ending	WP$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	wh-adverb

BP chunking

The set of chunk labels is the same as for the Arabic corpus, also following the IOB tagging scheme.

Final annotated text

That is the whole process we need to do for English. So, for the final version of the corpus we just compile all the information and write it in the format for the Moses decoder:

```
france|france|NN|B-NP ,|,|,|I-NP britain|britain|NN|I-NP ,|,|,|O
italy|italy|RB|B-ADVP ,|,|,|O germany|germany|NN|B-NP ,|,|,|O
ireland|ireland|NN|B-NP ,|,|,|O spain|spain|NN|B-NP ,|,|,|O
and|and|CC|O luxembourg|luxembourg|NN|B-NP came|come|VBD|B-VP
first.|first.|RB|B-ADVP a|a|DT|B-NP part|part|NN|I-NP
from|from|IN|B-PP the|the|DT|B-NP european|european|JJ|I-NP
companies|company|NNS|I-NP ,|,|,|O the|the|DT|B-NP
issued|issue|VBN|I-NP capital|capital|NN|I-NP of|of|IN|B-PP
companies|company|NNS|B-NP operating|operate|VBG|B-VP in|in|IN|B-PP
egypt|egypt|NN|B-NP reached|reach|VBN|B-VP le126|le126|NN|B-NP
billion|billion|CD|I-NP up|up|RP|B-ADVP till|till|IN|B-PP
december|december|NN|B-NP 2000.|2000.|CD|I-NP such|such|JJ|I-NP
capital|capital|NN|I-NP is|be|VBZ|B-VP of|of|IN|B-PP
10,000|10,000|CD|B-NP investment|investment|NN|I-NP
companies|company|NNS|I-NP set|set|VBN|B-VP up|up|RP|B-PRT
under|under|IN|B-PP the|the|DT|B-NP investment|investment|NN|I-NP
law|law|NN|I-NP .|.|.|O
```

3.4 Bare SMT System

Once the data have been prepared and before starting the training process we calculate the language model using the SRILM Toolkit [40]. For words we build the 5-gram language model by interpolated Kneser-Ney discounting. For linguistic factors such as lemma, part-of-speech and chunk label we generate 5-gram models without applying any discounting.

As for the translation model, we need to obtain the word alignments before calculating the probability tables. We use the `GIZA++` Toolkit [35] for that purpose. This software implements the IBM models but here it is only used to obtain the alignments in the two directions of translation. The word classes demanded by `GIZA++` are calculated with the `mkcls` program [32] by Franz Josef Och as well. The final alignment is obtained by applying the *grow-diag-final* heuristic (see Section 2.1.2).

From these alignments the maximum likelihood lexical translation tables in both directions are estimated. On the other hand, all the phrases compatible with the alignment are extracted and the phrase translation probabilities, again in both directions, estimated. All these steps are done with the training script provided with the `Moses` distribution.

As we have said we use the `Moses` decoder [26, 24]. The decoder implements a beam search where the output sentence is generated from left to right in form of hypotheses. Among all the hypothesis, that with the lowest cost (or highest probability) is selected as best translation.

Finally, we optimise the weights of every probability table by optimising translation performance on a development set. That sums up to 8 weights λ_i: 1 corresponding to the language model λ_{lm}, 2 for the two directions of the lexical translation tables λ_{lg} and λ_{ld}, 2 for the two directions of the phrase translation tables λ_g and λ_d, the distortion model λ_{di}, and the phrase and word penalties λ_{ph} and λ_w. For this optimisation we use a minimum error rate training (MERT) [33] where BLEU is the reference score.

3.5 Hybrid MT System

The hybrid system is obtained by adding a machine learning component to the bare SMT system. Language models and the MLT translation models are estimated in the same way, but now we use the methodology in Giménez and Màrquez [19] to estimate the discriminative phrase translation model.

For every selected phrase, we use linear SVMs and train the classifier for every possible translation phrase as explained in Section 2.2.1. For that we use the SVM^{light} package[6] [21]. This stage gives a SVM score for each instance of a phrase, and that score is converted into a probability using the softmax function as defined in Ref. [4]. Since this is done for every instance of a phrase, the probability tables would be enormous, and before calculating them it is convenient to filtrate for only the phrases appearing in the test.

This new translation table is added logarithmically to the full model as shown in Eq. 2.7. That allows to use a standard decoder as Moses with the only modification of a new score in the translation model. The optimisation process is again the same, a minimum error rate training is applied but now 9 weights must be fit: the 8 from the bare SMT system plus $\lambda_{\textsc{dpt}}$.

[6] http://svmlight.joachims.org/

Chapter 4

Experiments and evaluation

The following issue is to evaluate the systems described in the previous chapter. Some of the experiments are addressed to explore the effects of the preprocessing in the translation and others those of linguistic factors. Finally, we evaluate the improvements given by a discriminative phrase selection.

4.1 Word segmentation of Arabic

The first experiment is devoted to study the impact of word segmentation in Arabic. For this, we use the three data sets introduced in Section 3.3.1 with three different levels of tokenization. With the coarser tokenization the sparsity of the vocabulary increases and the mean length of an Arabic sentence is 0.80 times the English one. The first level of clitic segmentation diminishes the sparsity and equals the ratio between lengths to 0.92. With the second level, Arabic sentences are already longer than the English ones with ratio 1.11. In all these cases we use a language model computed from each training set without adding data out of domain.

We see in Table 4.1 that the best results are obtained when the sentence length in both languages is comparable, where only punctuation marks and all the clitics except *Al* are segmented. The additional separation of the determiner worsens the BLEU score by several possible reasons. First, because the method used to segment out *Al* can be segmenting true full words. Second, because Arabic has some determiners which have no analogy in English such as those before adjectives that are added when the noun is determined as well. Finally, the difference in the sentence length can make worse the quality of the alignments.

	Arabic→English		English→Arabic	
	dev	test	dev	test
punct.	25.76	23.46	23.50	16.17
punct.+clitics	26.25	23.81	26.54	19.67
punct.+clitics+Al	25.28	23.21	32.46	26.68

Table 4.1: BLEU scores for the translation of the NIST's news compilation with three different levels of segmentation (see text).

In fact, El Isbihani et al. [15] tested different segmentation methods and obtain the best results for the segmentation obtained with `ASVMTools`, that is without separating *Al-*, for a corpus built from the corpora of the Arabic-English NIST task. The worst results in their case correspond to the method that most segmentates the corpus with a ratio between the mean Arabic sentence length and the English one of 1.20.

With these results in mind, we use in the following the Arabic part of the corpus with the clitic segmentation of `ASVMTools`. At this point it is worth noticing that in this work segmentation is useful for the Arabic-to-English translation. In order to be used for the English-to-Arabic direction, on would need an algorithm to join the clitics again. This is not a trivial step and should be learned independently. Since the aim of the work has been the Arabic-to-English task of the NIST 2008, we postpone this issue for the future. For completeness, Table 4.1 shows the BLEU scores in this direction of translation too, but higher values must be attributed to the level of segmentation of the sentences: the more segmented a phrase is, the higher the number of correct words that involves a correct translation.

4.2 SMT system: combination of models with linguistic information

In this section, we describe a naïve way to include linguistic information within a statistical framework. For that purpose we use the small training set of the United

Nations corpus with 20,000 lines. The language model is estimated from the same training set.

In the first approach, each word of the corpus is concatenated with its lemma (l), part-of-speech (p) or chunk labels (c). For example, the word *books* can be converted into just one token if we include the lemma (wl): books##book, two options appear for the addition of the part-of-speech (wp): books##NNS and books##VBZ, and the larger number of alternatives is given with the chunk labels (wc): books##B-NP, books##I-NP, books##B-VP, and so on. The main disadvantage of this method is that it increases the vocabulary size for a same corpus size:

	Words	Vocabulary			
		word	word##lemma	word##PoS	word##chunk
Arabic	805,458	28,356	–	33,912	38,988
English	642,386	19,612	19,951	22,535	29,622

We see that the inclusion of the lemma hardly increases the information, the vocabulary does not augment tremendously. However, the part-of-speech and above all the chunk label increase the vocabulary size and therefore the sparsity.

The upper part of Table 4.2 shows the BLEU score for a baseline indicated by w where the translation is done with the standard SMT system, and for the combinations wl, wp and wc inserted in the corpus. In general, the Arabic-to-English direction gets slightly better results than from English to Arabic.

The addition of the lemma into the English part of the corpus –remember that the Arabic one has not been annotated with lemmas– improves the BLEU score in both the Arabic-to-English direction and the opposite one. This is because the specification of lemmas in English can improve the word alignments, and therefore the global translation. The additional information into the language model contributes to the improvement as well. The part-of-speech improves the quality only in the English-to-Arabic translation. The increment of sparsity is more important in the English part, since the number of tags in Arabic is 24 and in English 36. Then, the possible improvement in the alignments which is given in both directions can be compensated by a too sparse English language model. As for the case where the chunk label is included, wc, the gain in information is widely compensated by the consequent augment of sparsity. That worsens the results in both directions, especially in the Arabic-to-English direction where again the language model reflects the effect of a larger vocabulary.

	Arabic→English		English→Arabic	
	dev	test	dev	test
w (baseline)	24.70	23.82	26.83	22.85
wl	24.74	24.28	26.95	23.34
wp	23.93	23.18	27.00	23.12
wc	22.93	22.07	25.97	22.25
wl_{fac}	25.06	24.24	26.74	22.72
wp_{fac}	24.70	23.69	27.00	22.97
wc_{fac}	23.79	22.97	27.04	23.05
w+l	24.40	23.41	25.53	22.13
w+p	23.07	22.14	25.98	22.06
w+c	23.08	22.03	23.08	19.62
w+wl	24.86	23.98	27.17	23.01
w+wp	24.52	23.53	27.18	22.97
w+wc	23.77	22.69	26.64	22.75
w+wl+wp	23.90	23.74	23.08	19.40

Table 4.2: BLEU scores obtained with a training set of 20,000 lines of the United Nations corpus. Linguistic information is added to the baseline w by modifications of the token (wl, wp and wc) or by the addition of translation tables ($w + x$).

As a second experiment, we combine two translation models as extra components in a log-linear model. For this, we use the translation table corresponding to the translation of the words as a single factor with one corresponding to the linguistic features or combinations of them. We use the sum symbol + to mark this kind of systems. None of the combinations surpasses the best result given by a direct concatenation of the word with a feature (Table 4.2).

The combination of one translation table corresponding to the direct translation of tokens and another one corresponding to one linguistic factor ($w + l$, $w + p$ and $w + c$) is one BLEU point below that obtained for the concatenation of the word with the feature. The Moses decoder allows for a similar combination with factored models. In that case there are two language models too, one for each feature, but

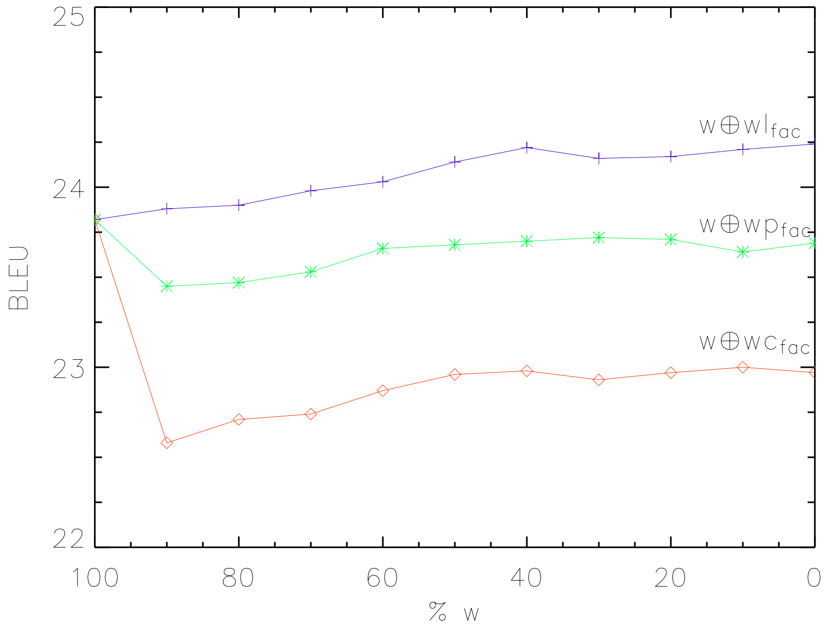


Figure 4.1: BLEU score for the combination of models $w{\oplus}wl_{fac}$, $w{\oplus}wp_{fac}$ and $w{\oplus}wc_{fac}$ by a global weight of every individual model. None of the combinations improves the individual scores.

an only translation table with two factors in a similar way we did in the previous experiment. We indicate this case with the subindex *fac*. The use of one translation table instead of two diminishes by five the number of weights to fit in the tuning process, and that eases the finding of the absolute minimum. With this, results improve our combinations $w + l$, $w + p$ and $w + c$, but still the concatenations wl, wp and wc reach better translations under the point of view of the BLEU score. Table 4.2 also shows other combinations of translation tables, each one with each correspondent language model, but none of the combinations is better than the use of the lemma alone wl. With these results, one would expect that the lemmatization of Arabic is going to help in the translation too.

We have just explained that the combination of two translation tables increases the number of weights and therefore makes harder the tuning process. As a final check and focusing again into the Arabic-to-English translation, we combine couples of translation tables giving a global weight to each of them with the already optimised λ's. We indicate this direct sum of translation tables with the symbol $\oplus$. Table 4.2 shows and example of this for a case where the translation is done giving different percentages to the w and wc_{fac} translation tables.

$\%w \oplus \%wc_{fac}$	*BLEU*				Translation table w					Translation table wc_{fac}					
		λ_{di}	λ_{lm1_w}	λ_{lm_wc}	λ_g	λ_{lg}	λ_d	λ_{ld}	λ_{ph}	λ_g	λ_{lg}	λ_d	λ_{ld}	λ_{ph}	λ_w
$100 \oplus 0$	23.82	0.0793	0.2229	–	0.0996	0.0831	0.1176	0.0858	−0.0755	–	–	–	–	–	−0.2359
$90 \oplus 10$	22.58	0.0826	0.2278	0.0036	0.0897	0.0747	0.1059	0.0772	−0.0679	0.0014	0.0161	0.0142	0.0034	−0.0011	−0.2341
$80 \oplus 20$	22.71	0.0860	0.2327	0.0071	0.0797	0.0664	0.0941	0.0687	−0.0604	0.0027	0.0322	0.0283	0.0068	−0.0022	−0.2323
$70 \oplus 30$	22.74	0.0893	0.2376	0.0107	0.0697	0.0581	0.0824	0.0601	−0.0528	0.0041	0.0483	0.0425	0.0102	−0.0033	−0.2306
$60 \oplus 40$	22.87	0.0926	0.2425	0.0143	0.0598	0.0498	0.0706	0.0515	−0.0453	0.0055	0.0644	0.0567	0.0136	−0.0044	−0.2288
$50 \oplus 50$	22.96	0.0960	0.2474	0.0178	0.0498	0.0415	0.0588	0.0429	−0.0377	0.0068	0.0805	0.0709	0.0169	−0.0055	−0.2270
$40 \oplus 60$	22.98	0.0993	0.2523	0.0214	0.0398	0.0332	0.0470	0.0343	−0.0302	0.0082	0.0966	0.0850	0.0204	−0.0066	−0.2252
$30 \oplus 70$	22.93	0.1026	0.2572	0.0250	0.0299	0.0249	0.0353	0.0257	−0.0226	0.0096	0.1127	0.0992	0.0238	−0.0077	−0.2235
$20 \oplus 80$	22.97	0.1060	0.2621	0.0285	0.0199	0.0166	0.0235	0.0172	−0.0151	0.0109	0.1288	0.1134	0.0272	−0.0088	−0.2217
$10 \oplus 90$	23.00	0.1093	0.2670	0.0321	0.0099	0.0083	0.0118	0.0086	−0.0076	0.0124	0.1449	0.1276	0.0306	−0.0099	−0.2199
$0 \oplus 100$	22.97	0.1126	0.2719	0.0356	–	–	–	–	–	0.0137	0.1611	0.1418	0.0339	−0.0110	−0.2181
$w + wc$	22.69	0.0724	0.1753	0.0418	0.0254	0.0070	0.0117	0.0567	−0.1045	0.0306	0.0646	0.0313	0.0447	−0.0767	−0.2573

Weights

λ_{di}: distortion, $\lambda_{lm\_w}, \lambda_{lm\_wc}$: language models,

λ_d: phrase translation probability $\phi(e|f)$, λ_{ld}: lexical weighting $lex(e|f)$, λ_g: phrase translation probability $\phi(f|e)$, λ_{lg}: lexical weighting $lex(f|e)$,

λ_{ph}: phrase penalty, λ_w : word penalty.

We have done this analysis with three combinations, $w \oplus wl_{fac}$, $w \oplus wp_{fac}$ and $w \oplus wc_{fac}$, and show the corresponding BLEU scores graphically as a function of the proportion of each component in Figure 4.1. One can see that the combination of both sources improves the result of the lowest translation table but hurts the score reached by the most informative feature or combinations of features alone.

4.3 Hybrid system: discriminative phrase translation

Next, we analyse in more detail the steps and results obtained with the hybrid system. The whole training is done using the news compilation corpus with 123,662 lines.

4.3.1 Phrase extraction

Since the system is a phrase-based translation system, the phrase extraction step is important for the final result. A larger number of phrases gives more translation options available to the decoder, and therefore it is usually better recall in front of precision in what refers to the quality of the extracted phrases. So, phrase alignments obtained by the intersection of words alignments produce in general better translation results than the union, which, on the other hand, leads to the subset of more precise phrases.

Here, we use two different heuristics to extract the phrases. For the extraction done with the `MLT` package we apply the heuristic *diag-and* as explained in Refs. [25, 36]. The heuristic *grow-diag-final* is used with the `Moses` software. Both of the heuristics complement the intersection points with some points belonging to the union, but the second one generates more phrases due to an additional final step that adds some extra alignment points. Table 4.3 shows the number of phrases extracted by the two methods according to the number of occurrences of the phrase in the corpus. Besides the different heuristics, we further increase the number of phrases corresponding to those extracted with the `Moses` software by considering phrases up to a length of 7 words instead of 5 words as with `MLT`. The distribution is seen in Table 4.4. The proportions through partitions are the same for both

	MLT		Moses	
Occurrences	phrases	%	phrases	%
2-5	239617	74.2	449660	76.8
6-10	43228	13.4	80261	13.7
11-50	30806	9.5	45066	7.7
51-100	4360	1.4	5009	0.9
101-500	3937	1.2	4299	0.7
501-1000	521	0.2	565	0.1
1001-10000	378	0.1	421	0.07
> 10000	22	0.007	26	0.004
Total:	322869	100	585307	100

Table 4.3: Number of phrases extracted according to the number of occurrences in the corpus, for both the MLT and Moses systems.

heuristics but we obtain a higher number of phrases with *grow-diag-final*. With these differences we end up with two sets of phrases. We call MLT set the small set with 322,869 phrases and Moses set the larger one with 585,307 phrases.

All of these phrases will be used to construct the translation tables by frequency counts, but we consider only those appearing more than 100 times in the corpus to be representative enough to train the classifiers. That represents about 1% of the total amount of phrases, but since they are the most frequent ones they will cover most of the test set if it belongs to the same domain.

4.3.2 Discriminative phrase selection

Before approaching the full task of translation we show some details of the subtask of phrase selection. The strength of this method is its capability of using the context of each phrase and the linguistic information available in order to select the best translation. This is especially useful to solve ambiguities, a very common semantic phenomenon in Arabic.

As an archetypical example we comment the different meaning of the word transliterated as *Elm*. Due to the non-vocalization of written texts, one can find *Elm* meaning “science” (*Eilom*), “flag” (*Ealam*) or “to know” (*Ealim*). These three

	MLT		Moses	
Length	phrases	%	phrases	%
1	30971	9.6	29949	5.1
2	102710	31.8	127878	21.8
3	94084	29.1	132874	22.7
4	62389	19.3	107589	18.4
5	32715	10.1	84527	14.4
6	-	-	61775	10.6
7	-	-	40718	7.0
Total:	322869	100	585307	100

Table 4.4: Number of phrases extracted according to its length, for both the MLT and Moses systems.

words are perfectly distinguishable when speaking but not when reading. The same happens with *ktb*, a word that can be read as *katab* (“to write”), *kitab* (“book”) or *katib* (“writer”). This kind of ambiguity is to be added to homonyms. Besides, verbal declinations can further increase the number of meanings.

We have trained linear SVMs to solve this problem. The features for training the classifier are extracted from both the source phrase and source sentence in Arabic but not from the target in English. From the phrase we consider word, part-of-speech, coarse part-of-speech and chunk labels n-grams. The same features are extracted from the full sentence with the addition of the bag-of-words which keeps the words at the right and at the left of the phrase.

The word *Elm* is found in the corpus together with the article: *AlElm*. This token is seen in 114 examples with 10 possible translations, being the most frequents:

AlElm:

Translations	flag	science	knowledge	mind	the flag
# examples	47	26	15	9	6

We extract the features for each of the examples that occur as translation at least a 0.5% of the times. In a case like this with 114 examples, all translations are considered. For instance, for one example where *AlElm* is translated as “knowledge”:

Sentence:

w tAbE mr\$d AlIxwAn “ In AlElm AlmTlwb fy dyn nA hw kl Elm nAfE tbqY l AlnAs vmrt h , swA’ kAn ElmAF \$rEyAF Ow ElmAF tjrybyAF .

Phrase features:

word *n*-grams	AlElm
PoS *n*-grams	NN
coarse PoS *n*-grams	N
chunk *n*-grams	B-NP

Sentence features:

word	(AlmTlwb)$_{1}$, (fy)$_{2}$, (dyn)$_{3}$, (nA)$_{4}$, (hw)$_{5}$,
n-grams	(” In)$_{-2}$, (AlIxwAn)$_{-3}$, (mr\$d)$_{-4}$, (tAbE)$_{-5}$,
	(AlmTlwb fy)$_{1}$, (fy dyn)$_{2}$, (dyn nA)$_{3}$, (nA hw)$_{4}$,
	(In AlmTlwb)$_{-1}$, (AlIxwAn ”)$_{-3}$, (mr\$d AlIxwAn)$_{-4}$, (tAbEmr\$d)$_{-5}$
	(AlmTlwb fy dyn)$_{1}$, (fy dyn nA)$_{2}$, (dyn nA hw)$_{3}$,
	(In AlmTlwb fy)$_{-1}$, (” In AlmTlwb)$_{-2}$, (AlIxwAn ” In)$_{-3}$,
	(mr\$d AlIxwAn ”)$_{-4}$, (tAbE mr\$d AlIxwAn)$_{-5}$
PoS	(JJ)$_{1}$, (IN)$_{2}$, (NN)$_{3}$, (PRP\$)$_{4}$, (PRP)$_{5}$,
n-grams	(PUNC IN)$_{-2}$, (NN)$_{-3}$, (NN)$_{-4}$, (VBD)$_{-5}$
	(JJ IN)$_{1}$, (IN NN)$_{2}$, (NN PRP\$)$_{3}$, (PRP\$ PRP)$_{4}$,
	(IN JJ)$_{-1}$, (NN PUNC)$_{-3}$, (NN NN)$_{-4}$, (VBD NN)$_{-5}$
	(JJ IN NN)$_{1}$, (IN NN PRP\$)$_{2}$, (NN PRP\$ PRP)$_{3}$,
	(IN JJ IN)$_{-1}$, (PUNC IN JJ)$_{-2}$,
	(NN PUNC IN)$_{-3}$, (NN NN PUNC)$_{-4}$, (VBD NN NN)$_{-5}$,
coarse PoS	(J)$_{1}$, (I)$_{2}$, (N)$_{3}$, (P)$_{4}$, (P)$_{5}$, (P I)$_{-2}$, (N)$_{-3}$, (N)$_{-4}$, (V)$_{-5}$
n-grams	(J I)$_{1}$, (I N)$_{2}$, (N P)$_{3}$, (P P)$_{4}$, (I J)$_{-1}$, (N P)$_{-3}$, (N N)$_{-4}$, (V N)$_{-5}$
	(J I N)$_{1}$, (I N P)$_{2}$, (N P P)$_{3}$,
	(I J I)$_{-1}$, (P I J)$_{-2}$, (N P I)$_{-3}$, (N N P)$_{-4}$, (V N N)$_{-5}$
chunk	(I-NP)$_{1}$, (B-PP)$_{2}$, (B-NP)$_{3}$, (I-NP)$_{4}$, (B-NP)$_{5}$,
n-grams	(O B-SBAR)$_{-2}$, (B-NP)$_{-3}$, (B-NP)$_{-4}$, (B-VP)$_{-5}$
	(I-NP B-PP)$_{1}$, (B-PP B-NP)$_{2}$, (B-NP I-NP)$_{3}$, (I-NP B-NP)$_{4}$,
	(B-SBAR I-NP)$_{-1}$, (B-NP O)$_{-3}$, (B-NP B-NP)$_{-4}$, (B-VP B-NP)$_{-5}$
	(I-NP B-PP B-NP)$_{1}$, (B-PP B-NP I-NP)$_{2}$, (B-NP I-NP B-NP)$_{3}$,
	(B-SBAR I-NP B-PP)$_{-1}$, (O B-SBAR I-NP)$_{-2}$, (B-NP O B-SBAR)$_{-3}$,
	(B-NP B-NP O)$_{-4}$, (B-VP B-NP B-NP)$_{-5}$
bag-of-words	left: AlIxwAn, mr\$d, tAbE
	right: \$rEyAF, AlmTlwb, AlnAs, Elm, ElmAF, dyn, kAn, kl, nAfE, swA’, tbqY, tjrybyAF, vmrt

	MLT phrases			Moses phrases		
Occurrences	#	Acc.DPT (%)	Acc.MFT (%)	#	Acc.DPT (%)	Acc.MFT (%)
100-500	3952	68.8	62.0	4310	66.5	58.7
501-1000	521	70.3	63.5	565	68.8	62.3
1001-5000	346	76.2	69.2	393	73.0	66.7
5001-10000	31	77.3	69.0	27	79.5	72.2
10001-50000	15	75.1	66.7	19	74.8	66.6
> 50000	7	75.4	65.8	7	80.7	76.2
Total:	4872	69.6	62.8	5321	67.3	59.8

Table 4.5: Mean accuracy obtained in the phrase translation task by the most frequent translation (MFT) and with SVMs (DPT) for two sets of phrases. Results are given for subsets of phrases grouped according to its frequency.

Since this phrase is an only word the phrase features are just unigrams. As for the sentence, one considers up to trigrams of features for tokens ranging from the position of the phrase minus five to the position plus five.

Training the classifier with the help of the previous features, we obtain, after a 10-fold cross-validation, an accuracy of 71.3%. The most frequent translation does it well the 49.6% of times. That is to say, one gets a 40% of relative improvement on the selection of the phrase translation. In general, the accuracy in the translation of phrases is improved with respect to that corresponding to the most frequent translation, but the amount of improvement depends on the phrase, the number of translations and the number of examples.

Table 4.5 shows the comparison of the accuracy obtained by SVMs, the Discriminative Phrase Translation (DPT), and that given by the Most Frequent Translation (MFT) for both the set of phrases extracted with MLT and Moses. The most frequent phrases of the MLT set get a larger improvement, but the low frequency ones improves one point more in the Moses set than in the MLT set. This is why, globally, the Moses set gets more benefits from the SVM classification. An increment of 7.5% in accuracy is obtained in this case for DPT.

4.3.3 Full translation

Finally, we integrate DPT predictions into the SMT system. To do this, we calculate the DPT predictions for all possible translations of all source phrases appearing in the test (or development) set. The input text is transformed by introducing identifiers in order to distinguish every distinct instance of every distinct phrase. These identifiers correspond to the number of occurrences of the word seen in the test set before the current one. For instance, the second time[1] the transliterated word *AlElm* appears in the set is annotated as $AlElm_1$:

$wywm\ \ AlAHd_8\ \ ,_{371}\ \ \$hdt_3\ \ Edp_8\ \ mdn_1\ \ AfgAnyp\ \ tZAhrAt\ \ AHtjAj\ \ ElY_{456}$
$Alrswm_{39}\ \ Alms\}yp\ \ l_{873}\ \ Alnby\ \ (_{186}\ \ S\ \ )_{186}\ \ ,_{372}\ \ Hyv_{28}\ \ tm_{22}\ \ AHrAq\ \ AlElm_1$
$AldnmArky\ \ ._{1128}$

For those words without subindex there is not DPT prediction.

In a similar way, translation tables must be modified. Now, each occurrence of every source phrase has a distinct list of phrase translation candidates with their DPT predictions. DPT predictions are only estimated for the phrases appearing in the test set. Still, indexing increments tremendously the size of the translation table, and, even when filtered for only the phrases in the test set, the resulting tables become larger than 1GB and do not fit into memory in decoding time. Therefore, we only keep the first 50 translations[2] for every phrase. Translations were ordered according to the discriminative probability or by weighting all the scores. This second method showed to be most robust with respect to the ordering done without the DPT prediction, although this way its addition changes the phrases to be kept for decoding.

Table 4.6 shows all the translations available for the phrase *AlElm* the second time it appears in the test set. In this case, the chosen translation would be the same both according to $P_{DPT}(e|f)$ and to $P_{MLE}(e|f)$, but one can already see in the table that the distribution of the probability mass is different for both predictions and that can alter the best choice.

[1] Indexing begins at 0.

[2] Using more than 20 translations per phrase during decoding was found to provide no improvement when applied to our baseline with respect to the case where only 20 translations are available.

f_i	e_j	$P_{DPT}(e\|f)$	$P_{MLE}(f\|e)$	$lex(f\|e)$	$P_{MLE}(e\|f)$	$lex(e\|f)$
AlElm_1	flag	0.1986	0.6438	0.5417	0.3241	0.2826
AlElm_1	the	0.0419	0.0001	0.0001	0.0207	0.0217
AlElm_1	mind	0.0401	0.0608	0.0425	0.0620	0.0543
AlElm_1	the flag	0.0397	0.4000	0.5417	0.0414	0.0786
AlElm_1	flag during	0.0394	0.6667	0.5417	0.0138	0.0001
AlElm_1	knowledge	0.0392	0.0846	0.0798	0.1103	0.0924
AlElm_1	flag caused	0.0387	1.0000	0.5417	0.0138	0.0001
AlElm_1	science	0.0377	0.1529	0.1477	0.1793	0.1413
AlElm_1	education	0.0377	0.0018	0.0029	0.0138	0.0163
AlElm_1	in mind	0.0371	0.0571	0.0425	0.0138	0.0004

Table 4.6: Example of a fragment of the translation table indexed in order to take into account DPT predictions.

In case we do not have a DPT prediction for a phrase, we complete the translation table by using the MLE prediction. We realised that the normalization of the DPT scores is not equal to one anymore, and that could be damaging the final results. In the future, we are planning to do a discounting and complete the translation table by assigning a small probability to the undetermined DPT predictions.

Notice that we make available to the decoder several scores. Therefore, the decoder does not always use the DPT prediction as the best translation. DPT is competing with the MLE prediction and the remaining features shown in Equation 2.7. The weight of every score is determined during the tuning process. In our results, the DPT prediction always has a larger weight than the MLE one, being $\lambda_{DPT} \sim 3\lambda_{MLE}$. We checked another configuration as well, where the discriminative probabilities $P_{DPT}(e|f)$ replace $P_{MLE}(e|f)$ instead of being added as an additional feature. We denote by *DPT* this last system where the DPT prediction replaces the MLE one, and by DPT^+ the system where the DPT prediction is added.

In order to study the impact of DPT predictions we perform a deep analysis by using an heterogeneous set of metrics for evaluation. In previous sections, we only used lexical metrics to evaluate the quality of the translation. Here, we use the IQ_{MT} package [17], which provides a rich set of more than 500 metrics at different

Level	Metric	SMT	DPT	DPT$^+$
Lexical	1-PER	**0.5248**	0.5224	0.5221
	1-WER	**0.3166**	0.3075	0.3081
	1-TER	**0.3679**	0.3606	0.3613
	BLEU	0.2388	0.2387	**0.2396**
	NIST	**6.4044**	6.3263	6.3225
	GTM (e=1)	0.5708	**0.5730**	0.5705
	GTM (e=2)	**0.2166**	0.2154	0.2161
	GTM (e=3)	**0.1756**	0.1743	0.1750
	RG-L	0.5290	**0.5305**	0.5276
	RG-S⋆	**0.3442**	**0.3443**	0.3410
	RG-SU⋆	**0.3634**	**0.3635**	0.3604
	RG-W-1.2	0.3085	**0.3111**	0.3091
	MTR-exact	0.4948	**0.4991**	0.4974
	MTR-stem	0.5142	**0.5164**	0.5153
	MTR-wnstm	0.5183	**0.5207**	0.5193
	MTR-wnsyn	0.5396	**0.5430**	0.5413
Shallow Syntactic	SP-Op-⋆	0.4150	**0.4218**	0.4185
	SP-Oc-⋆	0.4193	**0.4237**	0.4214
	SP-NIST$_l$	**6.5745**	6.4771	6.4790
	SP-NIST$_p$	**5.6618**	5.6225	5.6161
	SP-NIST$_{iob}$	**4.7187**	4.6627	4.6795
	SP-NIST$_c$	**4.1460**	4.0858	4.1047
Syntactic	DP-O_l-⋆	0.2019	**0.2057**	0.2049
	DP-O_c-⋆	**0.3344**	0.3314	0.3318
	DP-O_r-⋆	**0.2347**	0.2319	0.2319
	DP-HWC$_w$	**0.0575**	0.0556	**0.0574**
	DP-HWC$_c$	0.2118	0.2168	**0.2181**
	DP-HWC$_r$	0.1422	0.1474	**0.1484**
	CP-O_p-⋆	0.4133	**0.4183**	0.4158
	CP-O_c-⋆	0.3823	**0.3868**	0.3847
	CP-STM	**0.2150**	**0.2144**	0.2128
Shallow Semantic	NE-M_e-⋆	0.2963	**0.2979**	0.2933
	NE-O_e-⋆	**0.3518**	**0.3515**	0.3472
	NE-O_e-⋆⋆	0.4161	**0.4217**	0.4185
	SR-M_r-⋆	**0.0868**	0.0841	0.0848
	SR-O_r-⋆	**0.2073**	0.2059	0.2048
	SR-O_r	**0.4143**	0.4076	0.4104
Semantic	DR-O_r-⋆	0.2101	**0.2192**	0.2157
	DR-O_{rp}-⋆	0.3139	**0.3272**	0.3204
	DR-STM	0.1563	0.1508	**0.1591**

Table 4.7: Automatic evaluation of MT results

linguistic levels[3]. We have selected a representative set of metrics, based on different similarity criteria:

- Lexical n-gram similarity (on word forms).
- Shallow-syntactic similarity (on part-of-speech tags and base phrase chunks).
- Syntactic similarity (on dependency and constituent trees).
- Shallow-semantic similarity (on named entities and semantic roles)
- Semantic similarity (on discourse representations).

A deeply detailed description of the metric set may be found in the IQ_{MT} technical manual [16].

Table 4.7 shows the results for the two systems with DPT prediction (*DPT* and *DPT*$^+$) together with a baseline where there is no DPT prediction (indicated by *SMT* in the table). In general, improvements are not significant for lexical metrics, except for the case of semantic metrics based on discourse representations and some syntactic metrics based on constituent and dependency parsing.

At the lexical level, while metrics based on rewarding longer n-gram matchings tend to prefer the *SMT* baseline, variants of ROUGE and METEOR tend to prefer the *DPT* system. Interestingly, the *DPT*$^+$ attains the highest score only according to BLEU, although not significantly.

At the shallow-syntactic level, metrics based on lexical overlapping over parts-of-speech and base chunk phrases prefer the *DPT* and *DPT*$^+$ alternatives, with a slight advantage in favour of the *DPT* system. However, NIST variants over sequences of lemmas, parts-of-speech, chunk labels and chunk types consistently prefer the *SMT* baseline.

At the properly syntactic level, metrics exhibit very different behaviours. For instance, with respect to metrics based on lexical overlapping over dependency trees, while the *'DP-*O_l*-*$\star$*'* metric (i.e., overlapping between lexical items hanging at the same level of the tree) gives a clear advantage to DPT systems, the *'DP-*O_c*-*$\star$*'* (i.e.,

[3] The IQ_{MT} software is available at `http://www.lsi.upc.edu/∼nlp/IQMT`.

lexical overlapping between grammatical categories) and *'DP-O_r-⋆'* (i.e., lexical overlapping between grammatical relations) metrics prefer the *SMT* baseline. In contrast, metrics based on head-word chain matching (HWC) over dependency trees and metrics based on lexical overlapping over constituent trees clearly prefer the DPT alternatives. Finally, the syntactic tree matching (STM) metric confers a similar score to the three systems.

At the shallow-semantic level, whereas metrics based on lexical overlapping and matching between named entities (NE) seem to prefer the *DPT* system, metrics based on semantic roles (SR) prefer the *SMT* baseline.

Finally, at the semantic level, metrics based on lexical overlapping between discourse representations (DR) confer a significant advantage to the DPT alternatives, specially in the case of the *DPT* system. The semantic tree matching (STM) metric gives a slight advantage to the DPT^+ system.

Chapter 5

Summary and conclusions

This work has been a first approach to the Arabic-to-English translation task. We have built a news training set from the compilation of six corpora supplied by the Linguistic Data Consortium (LDC) for the 2008 NIST Machine Translation Open evaluation. For complementary tests, we use parts of the United Nations corpus as well.

The final corpora have been enriched by annotating the sentences with linguistic information such as part-of-speech, chunk, and lemmas for the English part. This allowed us to include linguistic information within a standard SMT system.

As a first step, we explore the effects of the Arabic preprocessing in the translated output. Since Arabic is an agglutinative language, the level of segmentation is important to optimize the learning during the training process. The best results have been obtained for a clitic segmentation that do not separate the *Al-* determiner. This way, the source and target language have similar sentence lengths and the higher quality of the alignments due to that fact improves the BLEU score of the translation.

In a second part, we checked the impact of the inclusion of the linguistic information given by several methods. A direct concatenation of every word with its corresponding lemma gave the best translation results, despite only the English part of the corpus was annotated with lemmas. For the addition of parts-of-speech and chunks, the higher sparsity of the data compensated the increment in information, and we obtained no significant improvement.

The last part and our final proposal for the Arabic-to-English translation task for the 2008 NIST Machine Translation Open Evaluation corresponds to an SMT system that uses WSD techniques to select the best translation of a phrase given a source sentence. This method allowed us to take into account the context of each phrase. Phrase selection is treated here as a classification problem and linear SVMs are used to select the most adequate translation by using the context of the phrase and the linguistic information associated as features.

Although we get an increment of a 7.5% in accuracy for the subtask of phrase selection, the full translation task does not obtain significant improvements. Within the NIST 2008 evaluation context, our system has obtained a BLEU score (30.31) in the middle of the way of the best system (BLEU=45.57) and the worst one (BLEU=14.15). Nevertheless, most of the systems outperformed our results, being the mean BLEU score of 37.32.

However, we believe that the increment of 7.5% in accuracy in phrase selection is indicative of the possibilities of the method and we attribute the lack of improvement to a bad integration of the DPT predictions into the SMT system. According to this, we consider these results just preliminary results and propose several steps to improve the performance in a future work:

- Starting with the integration of the DPT predictions into the SMT system, we will further study different methods to complete the DPT probability scores in the translation table in the cases where there is not DPT estimation because of the lack of examples.

- Just as one uses both the MLE generative and discriminative translation model, $P_{\text{MLE}}(f|e)$ and $P_{\text{MLE}}(e|f)$, the discriminative learning in the Arabic-to-English and the English-to-Arabic directions would provide us with the equivalent probability scores for the DPT predictions: $P_{\text{DPT}}(f|e)$ and $P_{\text{DPT}}(e|f)$. We expect this additional feature to further improve the translation.

- Since other metrics can be more sensible to WSD than BLEU, the tuning of the λ parameters in the mert optimization with respect to the BLEU score is maybe not the best option. We will check other metrics. The optimization algorithm itself could be also substituted by another minimization method that explores more deeply the parameter space.

- Finally, a better preprocess of Arabic should help the training process. Up to now, we do not know of any free full Arabic lemmatizer, but given the positive impact of English lemmas in the second part of this work, the Arabic ones should be also important for the final result, especially in this Arabic-to-English direction. We plan to explore the effect of Arabic lemmas when added as a feature for the SVMs, together with the inclusion of other features.

Acknowledgments

This work has been done within the OpenMT project of the Spanish *Ministerio de Educación y Ciencia* in the *Grup de Processament del Llenguatge Natural* of the *Universitat Politècnica de Catalunya*.

I would like to thank the support of my supervisors, and, especially, I'm very grateful to Jesús Giménez for his help, motivation and enthusiasm.

Appendix A

Buckwalter transliteration

Arabic alphabet and the Buckwalter transliteration of each of the Arabic glyphs in its stand alone form. The Unicode symbol is given as well.

Name	Unicode name	Unicode	Buckwalter	Glyph
hamza-on-the-line	Arabic letter hamza	U+0621	’	ء
madda	Arabic letter aleph with madda above	U+0622	|	آ
hamza-on-’alif	Arabic letter aleph with hamza above	U+0623	>	أ
hamza-on-waaw	Arabic letter waw with hamza above	U+0624	&	ؤ
hamza-under-’alif	Arabic letter aleph with hamza below	U+0625	<	إ
hamza-on-yaa’	Arabic letter yeh with hamza above	U+0626	}	ئ
bare ’alif	Arabic letter alef	U+0627	A	ا
baa’	Arabic letter beh	U+0628	b	ب
taa’ marbuuTa	Arabic letter teh marbuta	U+0629	p	ة
taa’	Arabic letter teh	U+062A	t	ت
thaa’	Arabic letter theh	U+062B	v	ث
jiim	Arabic letter jeem	U+062C	j	ج
Haa’	Arabic letter hah	U+062D	H	ه

continued

Name	Unicode name	Unicode	Buckwalter	Glyph
khaa’	Arabic letter khah	U+062E	x	خ
daal	Arabic letter dal	U+062F	d	د
dhaal	Arabic letter thal	U+0630	*	ذ
raa’	Arabic letter reh	U+0631	r	ر
zaay	Arabic letter zain	U+0632	z	ز
siin	Arabic letter seen	U+0633	s	س
shiin	Arabic letter sheen	U+0634	$	ش
Saad	Arabic letter sad	U+0635	S	ص
Daad	Arabic letter dad	U+0636	D	ض
Taa’	Arabic letter tah	U+0637	T	ط
Zaa’ (DHaa’)	Arabic letter zah	U+0638	Z	ظ
cayn	Arabic letter ain	U+0639	E	ع
ghain	Arabic letter ghain	U+063A	g	غ
taTwiil	Arabic letter tatweel	U+0640	_	ـ
faa’	Arabic letter feh	U+0641	f	ف
qaaf	Arabic letter qaf	U+0642	q	ق
kaaf	Arabic letter kaf	U+0643	k	ك
laam	Arabic letter lam	U+0644	l	ل
miim	Arabic letter meem	U+0645	m	م
nuun	Arabic letter noon	U+0646	n	ن
haa’	Arabic letter heh	U+0647	h	ح
waaw	Arabic letter waw	U+0648	w	و
’alif maqSuura	Arabic letter alef maksura	U+0649	Y	ى
yaa’	Arabic letter yeh	U+064A	y	ي
fatHatayn	Arabic fathatan	U+064B	F	ً
Dammatayn	Arabic dammatan	U+064C	N	ٌ
kasratayn	Arabic kasratan	U+064D	K	ٍ
fatHa	Arabic fatha	U+064E	a	َ
Damma	Arabic damma	U+064F	u	ُ
kasra	Arabic kasra	U+0650	i	ِ
shaddah	Arabic shadda	U+0651	~	ّ
sukuun	Arabic sukun	U+0652	o	ْ
dagger ’alif	Arabic letter superscript alef	U+0670	‘	
waSla-on-alif	Arabic letter alef wasla	U+0671	{	

Bibliography

[1] Automatic Language Processing Advisory Committee (ALPAC). Language and Machines. Computers in Translation and Linguistics. Tech. Rep. Publication 1416, Division of Behavioural Sciences, National Academy of Sciences, National Research Council, Washington, D.C., 1966.

[2] Banerjee, S., and Lavie, A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization* (2005).

[3] Bangalore, S., Haffner, P., and Kanthak, S. Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)* (2007), pp. 152–159.

[4] Bishop, C. M. 6.4: Modeling conditional distributions. In *Neural Networks for Pattern Recognition* (1995), Oxford University Press, p. 215.

[5] Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics 19*, 2 (1993), 263–311.

[6] Cabezas, C., and Resnik, P. Using WSD Techniques for Lexical Selection in Statistical Machine Translation (CS-TR-4736/LAMP-TR-124/UMIACS-TR-2005-42). Tech. rep., University of Maryland, College Park. http://lampsrv01.umiacs.umd.edu/pubs/TechReports/LAMP_124/LAMP_124.pdf, 2005.

[7] Carpuat, M., Shen, Y., Xiaofeng, Y., and Wu, D. Toward Integrating Semantic Processing in Statistical Machine Translation. In *Proceedings of*

the International Workshop on Spoken Language Translation (IWSLT) (2006), pp. 37–44.

[8] Carpuat, M., and Wu, D. Evaluating the Word Sense Disambiguation Performance of Statistical Machine Translation. In *Proceedings of IJCNLP* (2005).

[9] Carpuat, M., and Wu, D. How Phrase Sense Disambiguation outperforms Word Sense Disambiguation for Statistical Machine Translation. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI)* (2007).

[10] Carpuat, M., and Wu, D. Improving Statistical Machine Translation Using Word Sense Disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2007), pp. 61–72.

[11] Chen, S. F., and Goodman, J. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language* (1999).

[12] Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B 39*, 1 (1977), 1–38.

[13] Diab, M., Hacioglu, K., and Jurafsky, D. Automatic tagging of arabic text: From raw text to base phrase chunks. In *5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04)* (Boston, MA., 2004).

[14] Doddington, G. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the 2nd Internation Conference on Human Language Technology* (2002), pp. 138–145.

[15] El Isbihani, A., Khadivi, S., Bender, O., and Ney, H. Morpho-syntactic arabic preprocessing for arabic to english statistical machine translation. In *Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics Annual Meeting (HLT-NAACL), Workshop on Statistical Machine Translation* (New York City, June 2006), Association for Computational Linguistics, pp. 15–22.

[16] Giménez, J. IQMT v 2.1. Technical Manual (LSI-07-29-R). Tech. rep., TALP Research Center. LSI Department. http://www.lsi.upc.edu/∼nlp/IQMT/IQMT.v2.1.pdf, 2007.

[17] Giménez, J., and Amigó, E. IQMT: A Framework for Automatic Machine Translation Evaluation. In *Proceedings of the 5th LREC* (2006), pp. 685–690.

[18] Giménez, J., and Màrquez, L. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th LREC* (2004).

[19] Giménez, J., and Màrquez, L. Context-aware Discriminative Phrase Selection for Statistical Machine Translation. In *Proceedings of the ACL Workshop on Statistical Machine Translation* (2007), pp. 159–166.

[20] Hutchins, W. J. Machine translation and machine-aided translation. *Journal of Documentation 34*, 2 (1978), 119–159.

[21] Joachims, T. Making large-scale support vector machine learning practical. 169–184.

[22] Kneser, R., and Ney, H. Improved backing-off for m-gram language modeling. *icassp 1* (1995), 181–184.

[23] Koehn, P., and Hoang, H. Factored Translation Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2007), pp. 868–876.

[24] Koehn, P., Hoang, H., Mayne, A. B., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computation Linguistics (ACL), Demonstration Session* (Jun 2007), pp. 177–180.

[25] Koehn, P., Och, F. J., and Marcu, D. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)* (Edomonton, Canada, May 27-June 1 2003).

[26] Koehn, P., Shen, W., Federico, M., Bertoldi, N., Callison-Burch, C., Cowan, B., Dyer, C., Hoang, H., Bojar, O., Zens, R., Constantin, A., Herbst, E., and Moran, C. Open Source Toolkit for Statistical Machine Translation. Tech. rep., Johns Hopkins University Summer Workshop. http://www.statmt.org/jhuws/, 2006.

[27] Kudo, T., and Matsumoto, Y. Fast methods for kernelbased text analysis. In *Proceedings of ACL-2003. Sapporo, Japan.* (2003).

[28] Lin, C.-Y., and Och, F. J. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)* (2004).

[29] Lita, L. V., Rogati, M., and Lavie, A. BLANC: Learning Evaluation Metrics for MT. In *Proceedings of the Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)* (2005), pp. 740–747.

[30] Melamed, I. D., Green, R., and Turian, J. P. Precision and Recall of Machine Translation. In *Proceedings of the Joint Conference on Human Language Technology and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)* (2003).

[31] Niessen, S., Och, F. J., Leusch, G., and Ney, H. An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation* (2000).

[32] Och, F. J. An efficient method for determining bilingual word classes. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics* (Morristown, NJ, USA, 1999), Association for Computational Linguistics, pp. 71–76.

[33] Och, F. J. Minimum error rate training in statistical machine translation. In *Proc. of the Association for Computational Linguistics* (Sapporo, Japan, July 6-7 2003).

[34] Och, F. J., and Ney, H. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)* (2002), pp. 295–302.

[35] Och, F. J., and Ney, H. A systematic comparison of various statistical alignment models. *Computational Linguistics 29*, 1 (2003), 19–51.

[36] Och, F. J., and Ney, H. The alignment template approach to statistical machine translation. *Computational Linguistics 30*, 4 (2004), 417–449.

[37] Och, F. J., Tillmann, C., and Ney, H. Improved alignment models for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (University of Maryland, College Park, MD, June 1999), pp. 20–28.

[38] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Association of Computational Linguistics* (2002), pp. 311–318.

[39] Slocum, J. A survey of machine translation: its history, current status, and future prospects. *Comput. Linguist. 11*, 1 (1985), 1–17.

[40] Stolcke, A. SRILM – An extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing* (2002).

[41] Tillmann, C., Vogel, S., Ney, H., Zubiaga, A., and Sawaf, H. Accelerated DP based Search for Statistical Translation. In *Proceedings of European Conference on Speech Communication and Technology* (1997).

[42] Vickrey, D., Biewald, L., Teyssier, M., and Koller, D. Word-Sense Disambiguation for Machine Translation. In *Proceedings of the Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)* (2005).

[43] Weaver, W. Translation. In *Machine Translation of Languages*, W. N. Locke and A. D. Boothe, Eds. MIT Press, Cambridge, MA, 1949/1955, pp. 15–23. Reprinted from a memorandum written by Weaver in 1949.

[44] Zens, R., and Ney, H. Improvements in phrase-based statistical machine translation. In *Proceedings of HLT-NAACL 2004* (Boston, MA, 2004), pp. 257–264.