Towards high-accuracy bilingual phrase acquisition from parallel corpora

Lionel Nicolas and **Egon W. Stemle** and **Klara Kranebitter**
{lionel.nicolas,egon.stemle,klara.kranebitter}@eurac.edu
Institute for Specialised Communication and Multilingualism
European Academy of Bozen/Bolzano

Abstract

We report on on-going work to derive translations of phrases from parallel corpora. We describe an unsupervised and knowledge-free greedy-style process relying on innovative strategies for choosing and discarding candidate translations. This process manages to acquire multiple translations combining phrases of equal or different sizes. The preliminary evaluation performed confirms both its potential and its interest.

1 Introduction

This paper reports on work in progress that aims at acquiring translations of phrases from parallel corpora in an unsupervised and knowledge-free fashion. The process described has two important features. First, it can acquire multiple translations for each phrase. Second, no restrictions is set on the size of the phrases covered by the translations, phrases can be of equal or different sizes. The process is a greedy-style one: it constructs a set of candidate translations and iteratively selects one and discards others. The iteration stops when no more candidate translations remain.

The main contributions of this paper are:

- a metric that evaluates a candidate translation by taking into account the likeliness in frequency of two phrases in a candidate translation (see sect. 6.3),
- a metric that evaluates a candidate translation by taking into account the number of occurrences of a candidate translation and the significance of each occurrence (see sect. 6.4),
- an approach that discards candidate translations by enforcing coherence with the ones validated in previous iterations.

This paper is organized as follows. In section 2 we introduce the terminology we use in this paper whereas in section 3 we describe the state of the art. Section 4 briefly introduces the research subjects for which the generated data is relevant for. Section 5 explains in an abstract fashion the ideas which implementations are later detailed in section 6. In section 7, we present and discuss a preliminary evaluation. Finally in section 8 and 9, we highlight possible future works and conclude.

2 Definitions

A *bitext* is composed of both source- and target-language versions of a sequences of tokens. The sequences are usually sentences, paragraph or documents. A *phrase* is a sequence of tokens. A *translation* is said to cover two phrases from two different languages when they are translation of one another. The *size of a phrase* corresponds to the number of tokens it contains. The *size of a translation* th is designated by $size(th)$ and corresponds to the sum of the sizes of the phrases it covers. A phrase *includes* another one if it includes all the tokens of the included phrase. A translation includes another one if the phrases it covers include the phrases covered by the included translation. An occurrence of a phrase in a bitext is called a *slot*. The value $slots(ph, b_n)$ is the number of slots of an phrase ph in a bitext b_n.

A candidate translation ct covering two phrases ph_i and ph_j is said to claim slots in a bitext b_n when $slots(ph_i, b_n) \neq 0$ and $slots(ph_j, b_n) \neq 0$. The number of slots claimed by ct is designated by the value $claims(ct, b_n)$ and is initially set to $min(slots(ph_i, b_n), slots(ph_j, b_n))$.A candidate translation ct is said to occur in a bitext b_n when $claims(ct, b_n) \neq 0$. Slots of a phrase ph_i in a bitext b_n are said to be *locked* when they cannot be claimed any more by any candidate translations. The number of locked slots of a phrase ph_i is designated by $locks(ph_i, b_n)$ and is initially set to 0.

3 Previous Work

The closest related work with fairly equivalent objectives, we found so far is the one of Lavecchia et al. (2008) where mutual information is used to extract translations of small phrases which quality is evaluated through the performance of a machine translation tool.

In a more indirect fashion, the method presented here can be related to phrase alignment and bilingual lexicon extraction.

Phrase alignment, a key aspect to improve machine translation tool performances, is for most methods such as Koehn et al. (2003), Zhang and Vogel (2005) or Deng and Byrne (2008) the task of acquiring a higher level of alignment from a corpus originally aligned on the word level. Even though it can allow to perform phrase translation extraction in a later stage, the two subjects are similar but not equivalent in the sense since input data and objectives are different. The evaluation protocol usually involves studying the performances of a machine translation tool such as Moses (Koehn et al., 2007) taking as input data the alignment.

Because word forms are the smallest type of phrases, the work presented is related to bilingual lexicon extraction. Many early approaches for deriving such lexicon from parallel corpora use association measures and thresholds (Gale and Church, 1991; Melamed, 1995; Wu and Xia, 1994). The association measures ar meant to rank candidate translations and the threshold allow to decide which one are kept or discarded. Although most association measures focus on recurrent occurrences of a candidate translation, other methods like Sahlgren and Karlgren (2005) and Widdows et al. (2002) use semantic similarity.

As it has been later explained later in Melamed (1997) and Melamed (2000), such strategy keeps many incorrect translations because of indirect associations, i.e, pairs of phrases that often co-occur in bitexts but are not mutual translations. Nevertheless, since translations tend be naturally more recurrent than the indirect associations, the counter-measure is generally to discard in a bitext a candidate translation if it covers a phrase covered by another one with a greater score (Moore, 2001; Melamed, 1997; Melamed, 2000; Tsuji and Kageura, 2004; Tufis, 2002).

In Tsuji and Kageura (2004), an extension of the method described in Melamed (2000) has been designed to cope with translations with very few occurrences.

4 Applicability

The extracted phrase translations can be used by tools or resources that deal directly or indirectly with translations.

The most direct application is to use such data as input for machine translation or memory-based translation systems. Another interesting use would be to exploit the phrase translations to directly perform phrase alignment.

Because word forms are the smallest type of phrases, the data can also be adapted and used for subjects that take advantage of bilingual lexicons. For example, the acquired translations could be used for extending multilingual resources such as Wordnets or help perform word sense disambiguation (Ng et al., 2003).

Because the process can cope with multiple translations, many homonyms or synonyms/paraphrases could also be derived (Bannard and Callison-Burch, 2005) by studying phrases that can be translated to a same phrase.

5 Design Goals

An abstract algorithm for extracting translations could be summarized by the following three objectives: (1) generate a set of candidate translations that includes the correct ones. (2) classify them and ensure that the correct ones are the best ones. (3) decide how many are to be kept.

The process we designed implements that abstract strategy in a greedy style way: it iterates

over a set of candidate translations and, at the end of each iteration, it validates one and discard others. The process finally stops when no candidate translations are left. The first objective thus remains the same. The second and third objectives are however final results: the classification of the best candidate translations and the number that are to be kept is only be established when the process stops. By being a greedy-style process, the task of deciding what are the correct translations is split into less-difficult sub-tasks, one for each iteration, where the process "just" needs to have as its best candidate translation a correct one.

5.1 Design criteria applied

The process should be able to acquire translations covering phrases of any sizes, i.e. strict restrictions on the size are to be avoided. The process should be able to acquire multiple (n to m) translations, i.e. strict restrictions on the number of translations for each phrase are to be avoided.

5.2 Abstract strategies for choosing

Local and global significance. All occurrences of a candidate translation should not have the same significance. The significance of an occurrence should take into account the number of other candidate translations also occurring in the same bitex with which it conflicts. In other words, the fewer are the candidate translations covering a same phrase in a bitext, the more interesting should be a candidate translation covering it. The process should also favour the candidate translations with a larger number of occurrences, i.e. the more recurrent a candidate translation is, the more interesting it is. The process should thus take into account both the number of occurrences and the significance of each, i.e. the significance of the occurrences of a candidate translation should be evaluated on "quality" and "quantity".

Frequencies likeliness Since we deal with translated texts, the vast majority of phrases in a bitext have a translation. Two phrases that can be translated one to the other should therefore have similar frequency. However, since the process should also cope with multiple translations, i.e. the process should also consider that occurrences can be divided among several translations.

5.3 Abstract strategy for discarding

The process should maintain coherence with previously validated candidate translations. Thus, previously validated ones should allow to discard the remaining ones that are not compatible with them. One can think of it as a sudoku-like strategy, i.e. taking a decision for one box/phrase allow to reduce the options for other boxes/phrases.

6 Detailed Description

6.1 Candidate generation

For both texts in each bitext, we generate the set of every phrases occurring and count how many times they occur, i.e. how many slots they have. We produce candidate translations by computing the Cartesian product between the two sets of phrases of every bitext and rule out most of them by applying the following permissive criteria:
(1) both covered phrases should occur at least min_occ times in the corpora,
(2) the covered phrases should co-occur in at least min_co_occ bitexts.
(3) both covered phrases covered should be among the max_cand phrases they co-occur the most with.

6.2 Choosing the best candidate

The process keeps at each iteration the candidate translation ct maximizing the following score:

$$size(ct) * like\_freq(ct) * significance(ct)$$

where $like\_freq(ct)$ is the evaluation of the likeliness of the frequencies of the phrases covered by ct and $significance(ct)$ is a score representing the significance of its occurrences (see below).

6.3 Evaluation of frequencies likeliness

As briefly sketched in 5.2, phrases covered by a correct candidate translation should have similar frequencies. In order to illustrate the idea, let's imagine that we only detect 1 to 1 translation and we classify phrases into three categories: low-frequency, medium-frequency, high-frequency. A metric trying to evaluate frequency likeliness will thus aim at giving a high score to candidate translation that cover phrases both classified in the same category and a lower one when classified in two different categories.

In practice, we do not classify the phrases into categories but assign to each phrase ph a frequency degree $fdeg(ph) \in [0, 1]$. This degree represents how frequent it is with regards to the other phrases of the same language. The most frequent ones receive a degree close to 1 and the less frequent ones a degree close to 0. We compute the frequency degree of a phrase ph as

$fdeg(ph) = \frac{(nb\_inf+1)}{nb\_phrase}$

where nb_inf is the number of phrases less frequent than ph and $nb\_phrases$ is the total number of phrases of the language. Then, for each candidate translation ct covering two phrases ph_i and ph_j, we compute a score

$$like\_freq(th) = 1 - abs(fdeg(ph_i) - fdeg(ph_j)).$$

Two aspects are to be considered. The first is the reason for computing the value $fdeg$ with the rank and not directly using the frequency. The reason is that it is not possible to know if a difference in x occurrences matters the same at different levels of frequency and with different languages. However, since we deal with translated texts, ordering them by frequency should stand from one language to the other and two phrases that are translations of one another should receive a similar $fdeg$.

The second aspect to consider is the fact of dealing with multiple translations. Therefore, occurrences can be divided among several translations. Since there is no reason for all translations to be equally balanced in frequency, one translation should dominate the others[1]. Every time a candidate translation is validated, we decrease the frequency of both covered phrases by the number of slots claimed by the validated candidate translation. If a phrase has multiple translations, validating the dominating one allows to "resynchronize" the frequency with the next dominating one. We thus recompute the $like\_freq$ value for the remaining candidate translations covering one of the two covered phrases.

6.4 Significance score

As briefly explained before in 5.2, we aim at evaluating the significance of a candidate translation on both the "quantity" and the "quality" of its occurrences.

For every phrase ph_i having slots available in a bitext b_n, we compute a value

$claimed(ph_i, b_n) = \sum_{k=0}^{n} claims(ct_k, b_n)$

of all ct_k candidate translations covering it. Then, for each candidate translation ct occurring in a bitext b_n and covering two phrases ph_i and ph_j, we compute a local score of significance

$local(ct_k, b_n) = \frac{claims(ct,b_n)}{claimed(ph_i,b_n)} * \frac{claims(ct,b_n)}{claimed(ph_j,b_n)}$

The value of $local(ct, b_n)$ will be equal to 1 if ct is the only one covering ph_i and ph_j and drop towards 0 as the number of candidate translations covering one of them raises.

Finally, we compute at every iteration the score

$significance(ct) = \sum_{i=0}^{n} local(ct_k, b_n)$

6.5 Updating candidate translations

We apply a strategy that maintain coherence with the previously candidate translations. For every occurrence of a validated candidate translation, two types of restrictions, strict and soft ones, are dynamically build so as to inflict handicap to the remaining candidate translations that are in conflict with the validated one.

Whenever a candidate translation ct conflicts with a restriction set in a bitext b_n its value $claims(ct, b_n)$ and thus its significance score and global score are re-evaluated for the next iteration. If the value $claims(ct, b_n)$ falls to 0, its occurrence is removed. If a candidate translation does not fulfil any more the original criteria of occurring in at least min_co_occ different bitext (see 6.1), it is discarded.

6.5.1 Strict restrictions

Strict restrictions lock the slots of the phrases covered by the previously validated candidate translations, i.e. some slots become not "claimable" any more. For two phrases ph_i and ph_j covered by a validated candidate translation ct and each bitext b_n in which it occurs, the values $locks(ph_i, b_n)$ and $locks(ph_j, b_n)$ are incremented by $claims(ct, b_n)$.

6.5.2 Soft restrictions

Soft restrictions impact candidate translation that cover phrases that are included or are included by a validated candidate translation. To

[1] especially if enforced for enhancing translation standardization.

each soft restriction $soft_m$ set on an phrase ph_i is associated a number of slots $num(soft_m)$ that a candidate translation cannot claim if it does not fulfil the condition.

Whenever a phrase PH_1 covered by a validated candidate translation ct includes a phrase ph_1, we consider that the translation of ph_1 should be included by the second n-gram PH_2 also covered by ct. We associate to such soft restriction $soft_m$ a $num(soft_m) = claims(ct, b_n)$. In other words, if PH_1 includes ph_1, we consider that for $claims(ct, b_n)$ slots of ph_1 its translation should be included in PH_2. For example if "la bella casa" in Italian is validated as the translation of "das schöne Haus" in German then, for any bitext containing both, phrases included in "la bella casa" should translate to phrases included in "das schöne Haus" and vice-versa.

Also, whenever a phrase PH_1 covered by a validated candidate translation ct is included in a phrase ph_1 and $slots(ph_1, b_n) = slots(PH_1, b_n)$, we consider that the translation of ph_1 should include the other phrase PH_2 covered by ct. We associate to such soft condition $soft_m$ a $num(soft_m) = claims(ct, b_n)$. In other words, if PH_1 is included in ph_1 and both phrases have the same original number of slots then PH_2 should be included by the translation of ph_1 at least $claims(ct, b_n)$ time(s). For example if "bella" is validated as the Italian translation of "schöne" in German then phrases including "bella" and having the same number of slots should translate into phrases including "schöne" and vice-versa.

6.5.3 Combining restrictions and updating the remaining candidate translations

Since we do not try to align phrases, combining the restrictions violated by a candidate translation must take into account that some restrictions may apply on slots that overlap between one another.

Regarding strict restrictions, we can ensure that two restrictions concern a set of slots that don't overlap even if we don't explicitly affect a given slot to a given strict restriction. For example, for a phrase ph_i with $m + n$ slots in a given bitext that is covered by two validated candidate translations ct_e and ct_h, we can tell that m slots have been locked by ct_e and n slots by ct_h and cannot be claimed by other candidate translations without stating explicitly which slot is locked by ct_e or ct_h.

Whenever a soft restriction is involved, simply adding the number of slots covered by the restrictions would be incorrect because we cannot establish if the restrictions violated do not overlap on a same set of slots. For example, let's consider a bitext containing both one occurrence of "la bella casa" in Italian and "das schöne Haus" in German with only one occurrence of "bella" and "schöne" in the whole bitext and two validated candidate translations ct_i and ct_j that associate "la bella" with "das schöne" and "bella casa" with "schöne Haus". A candidate translation ct_k that covers "bella" but does not associate it with "schöne" would violate both soft restrictions set by ct_i and ct_j. Simply adding the number of slots covered by the soft restrictions set by ct_i and ct_j would prohibit ct_k to claims two slots when only one is actually available. The same reasoning can be extended to phrases having more than one slot and to the combination soft and strict restrictions.

We thus look for the maximum number of slots that a remaining candidate translation ct occurring in a bitext b_n and covering two phrases ph_i and ph_j can claim. For each covered phrase ph, we compute a value $max\_soft(ph, ct, b_n)$ corresponding to the maximum of the $num(soft_m)$ values of the soft restrictions violated by ct for covering ph in b_n.

We then compute the value $sub\_claims(ph, ct, b_n)$ corresponding to the number of slots originally available $slots(ph, b_n)$ minus the maximum value between the number of slots locked by strict restrictions $locks(ph, b_n)$ and $max\_soft(ph, ct, b_n)$,

$$sub\_claims(ph, ct, b_n) = slots(ph, b_n) - max(locks(ph, b_n), max\_soft(ph, ct, b_n)).$$

Finally we update the $claims(ct, b_n)$ value in a similar manner as it has been first initialized

$$claims(th, b_n) = min(sub\_claims(ph_i, ct, b_n), sub\_claims(ph_j, ct, b_n))$$

It is important to note that generally only one slot is available for most phrases in a bitext. Therefore, conflicting with just one restriction in a bitext, be it a strict or a soft, is enough for most candidate translations to loose their occurrence.

7 Preliminary Evaluation

7.1 Input Corpora and configuration

To perform the evaluation, we extracted 50 000 bitexts from the Catex Corpus(Streiter et al., 2004). This bilingual corpus is a collection of Italian-language legal texts with the corresponding German translations. The bitexts in this corpus are composed of two sentences. The average length for the Italian sentences was $15,3$ tokens per sentence and $13,9$ for the German ones.

We have set the min_occ and min_co_occ variables to 3 and the max_cand variable to 20 (see Sect. 6.1). The maximum size of a candidate translation was set to 12 tokens, i.e 6 for each phrase covered. A total of 57 406 candidate translations have been generated.

7.2 Evaluation protocol

As explained in section 3, comparing the methods to the state-of-the-art is not straightforward. The closest method in terms of input data and objectives is the one described in Lavecchia et al. (2008). However, the results are evaluated according to the performance of a machine translation tool which is a task out of the reach of the preliminary evaluation we wanted to performed. We thus decided to establish an evaluation protocol as close as possible to the bilingual extraction methods such as those described in Melamed (2000) and Moore (2001). In these papers, the authors classify the precision of candidate translations into three categories: wrong, correct and "near misses". Even though these notions are quite straightforward for translations covering phrases of one or two tokens, they are more difficult to apply to larger ones. We thus report results obtained with several strategies for evaluating precision.

All evaluation strategies performed start from a manual evaluation that states the minimum number of tokens $errors(ct)$ that are to be added or deleted in both phrases covered by a candidate translation ct so as to obtain a fully valid translation. For each candidate translation, we thus start by evaluating how close it is from a perfect translation. For example, a candidate translation linking "landesgesetz vom 8 november" with "legge provinciale 8 novembre" is fully valid and receives a perfect score $errors(ct) = 0$ whereas another one linking "landesgesetz vom 8 november" with "provinciale 8 novembre" requires to add "legge" before "provinciale" and thus receives a score $errors(ct) = 1$. 6 samples of 500 candidate translations at different ranking have been manually evaluated by a trained interpreter.

We then applied the following two strategies to evaluate the precision of each candidate translations and compute average precisions over the 6 samples. The first strategy, called hereafter *Scalable precision*, assigns a precision score equal to $(size(ct) - errors(ct))/size(ct)$. The second strategy, called hereafter *Strict precision*, is a generic one that is instantiated with a threshold value $thresh$. It classifies a candidate translation ct as "correct" or "wrong" depending on whether or not $errors(ct)$ is under or above $thresh$: ct receives a precision score of 1 if $errors(ct) <= thresh$ and 0 otherwise. The thresholds chosen are not static but dynamically adjusted according to the size of the candidate translation evaluated. For example, if we set the threshold to $(3 * size)/12$ then a candidate translation ct_1 with $size(ct_1) = 6$ needs $errors(ct_1) <= 1.5$ to be considered correct whereas a candidate translation ct_2 with $size(ct_2) = 12$ needs $errors(ct_2) <= 3$.

7.3 Results

Table 1 provides some statistics about the validated candidate translations between two ranks: their average size, significance score and number of occurrences. Table 2 provides the results in terms of precision. The results are provided by evaluation criteria, sample and size of candidate translations. In each cell, the left value is the precision for the sample whereas the right one is a cumulative precision of all candidate translations with the same size. In table 3, each cell contains the number of candidate translations generated between two ranks according to their size and the proportion it represents among the ones generated between these two ranks.

As one can observe from table 2, precision remain stable and fairly high for the first two thirds. It then start decreasing noticeably and crumble in the last sample. An interesting result is that more than 50% of the candidate translations (30 000 over 57 406) have a close-to perfect precision

($> 98\%$ in scalable precision) and around 70% (40 500 over 57 406) have a reasonably high one ($> 95\%$ in scalable precision).

When analyzing the last part of the list we could observe that most errors are either random occurrence of a candidate translation covering frequent phrases or, as explained in Melamed (2000), indirect associations.

A first obvious observation is that the quality of the candidate translations does improve with their score, i.e. the higher the score is the better is the candidate translation.

When looking at table 1, apart from the first 10 000 candidate translations, we can see that the average frequency remains quite stable when compared with the average frequency, i.e. some less frequent candidate translations do receive a greater score than more frequent ones and therefore the average frequency remains stable. This behaviour meets our expectations since we wanted the process to not only consider the number of occurrences to decide whether or not a candidate translation was better than another one.

As said before, because the input corpora, the type of translations, and the form of evaluation are different, comparing our results to the state-of-the-art is challenging. Nevertheless, we noticed two aspects in favour of our results. For methods such as in Moore (2001) or Sahlgren and Karlgren (2005), the average number of occurrences of the candidate translations they reported is rather high. It is worth highlighting that our method achieves very high precision with frequent candidate translations. For other methods such as in Tufis (2002), the number of candidate translations seems relatively small when compared with the number of sentences provided. It is worth noting that we outputted more candidate translations than the total number of sentences we gave as input.

8 Future Work

By the very nature of the approach, the implementation process is heavy in terms of memory and computations. As it starts with an exponential number of phrases and thus an exponential number of candidate translations, running the process with the above configuration consumed up to 26 Gbyte during its first iterations, and ran for 5 days on a recent computer. For this technical reason, we had to limit our experiment and set the configuration to be more restrictive than originally intended. There is therefore a need to decrease the search space or dynamically adapt it.

Another evaluation using the generated translations in a machine translation system has been postponed as it implies external tools and additional knowledge. To do so, we could reproduce the evaluation performed in Lavecchia et al. (2008). As the method is very recent, we wanted first to have an preliminary overview of its potential to be able to consider further developments and evaluations. Another interesting evaluation would be to compare the phrases translations to the ones we could extract from the phrases alignment methods such as the one used in the Moses toolkit (Koehn et al., 2003).

As it has been designed as a greedy-style process, converting it into a beam-search process seems a viable option.

Like most natural language processing methods, this process would benefit from reducing the data sparsity. As it is currently designed, we could add a pre-processing that converts an input form-based parallel corpus into a lemmatised one.

Finally, several future works can be considered by reusing the data generated for other subjects that could take advantage of it (see sect. 4).

9 Conclusion

In this paper, we have presented an unsupervised and knowledge-free greedy-style process to derive multiple translations of phrases of equal or different sizes.

As it is still a recent and an on-going work, it has still much room for improvement. Several tracks towards this objective have been provided.

Finally, the preliminary evaluation performed has confirmed both its relevance and its potential.

Rank	Avg. size	Avg. signif	Avg. occ
$<$ 10000	6.99	44.67	26.02
10000-19999	7.18	6.50	8.35
20000-29999	5.48	4.38	7.06
30000-39999	4.61	3.02	6.38
40000-49999	3.77	1.45	7.66
$\geq$ 50000	2.24	0.23	5.67

Table 1: Statistics on average size, significance and occurrences depending on rank.

Scalable precision												
Rank \ Size	2	3	4	5	6	7	8	9	10	11	12	all
1-500	1.00/1.00	-	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	0.95/0.95	1.00-1.00
10001-10500	1.00/1.00	0.80/0.80	0.99/0.99	0.96/0.97	0.99/1.00	0.87/0.92	0.98/0.99	0.96/0.97	0.99/0.99	1.00/1.00	0.98/0.98	0.98-0.99
20001-20500	0.96/0.98	0.92/0.87	0.97/0.99	0.96/0.96	0.96/0.98	0.92/0.92	0.98/0.98	0.93/0.96	0.97/0.99	0.93/0.95	0.93/0.96	0.96-0.98
30001-30500	0.97/0.98	0.84/0.86	0.98/0.98	0.91/0.95	0.92/0.98	0.88/0.92	0.88/0.98	0.89/0.95	0.97/0.99	-	0.83/0.96	0.97-0.98
40001-40500	0.83/0.92	0.85/0.85	0.83/0.96	0.77/0.87	0.88/0.97	0.75/0.86	0.89/0.97	0.91/0.94	0.85/0.99	0.91/0.95	0.25/0.95	0.83-0.95
50001-50500	0.16/0.55	0.15/0.49	0.16/0.93	0.40/0.85	0.47/0.96	0.33/0.84	0.38/0.97	0.44/0.93	-	-	-	0.17-0.82
Strict precision, thresh = (0 * size) / 12												
Rank \ Size	2	3	4	5	6	7	8	9	10	11	12	all
1-500	1.00/1.00	-	0.99/0.99	1.00/1.00	0.99/0.99	1.00/1.00	1.00/1.00	1.00/1.00	0.95/0.95	1.00/1.00	0.43/0.43	0.98-0.98
10001-10500	1.00/1.00	0.40/0.40	0.97/0.99	0.82/0.88	0.97/0.98	0.42/0.65	0.93/0.96	0.71/0.81	0.95/0.95	1.00/1.00	0.85/0.76	0.92-0.95
20001-20500	0.95/0.98	0.75/0.62	0.92/0.96	0.78/0.82	0.85/0.94	0.56/0.61	0.88/0.93	0.43/0.66	0.80/0.93	0.68/0.76	0.50/0.68	0.84-0.91
30001-30500	0.96/0.97	0.53/0.57	0.95/0.96	0.65/0.77	0.67/0.92	0.33/0.57	0.75/0.93	0.50/0.65	0.67/0.93	-	0.00/0.67	0.90-0.91
40001-40500	0.79/0.90	0.68/0.65	0.66/0.92	0.33/0.58	0.57/0.87	0.23/0.45	0.45/0.88	0.46/0.60	0.25/0.92	0.00/0.71	0.00/0.65	0.62-0.85
50001-50500	0.14/0.54	0.12/0.38	0.07/0.88	0.40/0.57	0.33/0.86	0.00/0.43	0.00/0.87	0.00/0.59	-	-	-	0.13-0.73
Strict precision, thresh = (1 * size) / 12												
Rank \ Size	2	3	4	5	6	7	8	9	10	11	12	all
1-500	1.00/1.00	-	0.99/0.99	1.00/1.00	0.99/0.99	1.00/1.00	1.00/1.00	1.00/1.00	0.95/0.95	1.00/1.00	1.00/1.00	0.99-0.99
10001-10500	1.00/1.00	0.40/0.40	0.97/0.99	0.82/0.88	0.97/0.98	0.42/0.65	0.93/0.96	0.71/0.81	0.95/0.95	1.00/1.00	0.96/0.97	0.93-0.96
20001-20500	0.95/0.98	0.75/0.62	0.92/0.96	0.78/0.82	0.85/0.94	0.56/0.61	0.88/0.93	0.43/0.66	0.80/0.93	0.68/0.76	0.79/0.91	0.84-0.92
30001-30500	0.96/0.97	0.53/0.57	0.95/0.96	0.65/0.77	0.67/0.92	0.33/0.57	0.75/0.93	0.50/0.65	0.67/0.93	-	0.00/0.90	0.90-0.92
40001-40500	0.79/0.90	0.68/0.65	0.66/0.92	0.33/0.58	0.57/0.87	0.23/0.45	0.45/0.88	0.46/0.60	0.25/0.92	0.00/0.71	0.00/0.88	0.62-0.86
50001-50500	0.14/0.54	0.12/0.38	0.07/0.88	0.40/0.57	0.33/0.86	0.00/0.43	0.00/0.87	0.00/0.59	-	-	-	0.13-0.74
Strict precision, thresh = (2 * size) / 12												
Rank \ Size	2	3	4	5	6	7	8	9	10	11	12	all
1-500	1.00/1.00	-	0.99/0.99	1.00/1.00	0.99/0.99	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00-1.00
10001-10500	1.00/1.00	0.40/0.40	0.97/0.99	0.82/0.88	0.99/0.99	0.67/0.80	0.96/0.98	0.93/0.95	0.99/0.99	1.00/1.00	1.00/1.00	0.96-0.98
20001-20500	0.95/0.98	0.75/0.62	0.92/0.96	0.78/0.82	0.93/0.97	0.89/0.84	0.95/0.97	0.93/0.94	0.92/0.98	0.88/0.91	0.93/0.98	0.92-0.96
30001-30500	0.96/0.97	0.53/0.57	0.95/0.96	0.65/0.77	0.86/0.96	0.83/0.84	0.75/0.96	0.50/0.92	1.00/0.98	-	1.00/0.98	0.92-0.95
40001-40500	0.79/0.90	0.68/0.65	0.66/0.92	0.33/0.58	0.78/0.94	0.68/0.79	0.73/0.94	0.85/0.90	0.25/0.97	1.00/0.92	0.00/0.96	0.69-0.90
50001-50500	0.14/0.54	0.12/0.38	0.07/0.88	0.40/0.57	0.33/0.93	0.00/0.75	0.00/0.93	0.00/0.88	-	-	-	0.13-0.77
Strict precision, thresh = (3 * size) / 12												
Rank \ Size	2	3	4	5	6	7	8	9	10	11	12	all
1-500	1.00/1.00	-	0.99/0.99	1.00/1.00	0.99/0.99	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00-1.00
10001-10500	1.00/1.00	0.40/0.40	0.99/0.99	1.00/1.00	0.99/0.99	0.67/0.80	0.96/0.98	1.00/1.00	0.99/0.99	1.00/1.00	1.00/1.00	0.97-0.98
20001-20500	0.95/0.98	0.75/0.62	0.97/0.98	1.00/1.00	0.93/0.97	0.89/0.84	0.98/0.98	1.00/1.00	0.96/0.99	0.94/0.96	1.00/1.00	0.95-0.97
30001-30500	0.96/0.97	0.53/0.57	0.97/0.98	0.94/0.98	0.86/0.96	0.83/0.84	0.75/0.97	1.00/1.00	1.00/0.99	-	1.00/1.00	0.94-0.97
40001-40500	0.79/0.90	0.68/0.65	0.79/0.95	0.79/0.90	0.78/0.94	0.68/0.79	0.95/0.97	0.85/0.96	1.00/0.99	1.00/0.96	0.00/0.98	0.78-0.93
50001-50500	0.14/0.54	0.12/0.38	0.07/0.91	0.40/0.87	0.33/0.93	0.00/0.75	0.50/0.97	0.00/0.94	-	-	-	0.14-0.80
Strict precision, thresh = (4 * size) / 12												
Rank \ Size	2	3	4	5	6	7	8	9	10	11	12	all
1-500	1.00/1.00	-	0.99/0.99	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00-1.00
10001-10500	1.00/1.00	1.00/1.00	0.99/0.99	1.00/1.00	1.00/1.00	1.00/1.00	0.96/0.98	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	0.99-0.99
20001-20500	0.95/0.98	1.00/1.00	0.97/0.98	1.00/1.00	0.98/0.99	1.00/1.00	0.98/0.98	1.00/1.00	1.00/1.00	0.97/0.98	1.00/1.00	0.98-0.99
30001-30500	0.96/0.97	1.00/1.00	0.97/0.98	0.94/0.98	1.00/0.99	1.00/1.00	0.75/0.97	1.00/1.00	1.00/1.00	-	1.00/1.00	0.97-0.98
40001-40500	0.79/0.90	0.93/0.95	0.79/0.95	0.79/0.90	0.96/0.99	0.77/0.92	0.95/0.97	1.00/1.00	1.00/1.00	1.00/0.98	0.00/0.98	0.84-0.96
50001-50500	0.14/0.54	0.13/0.53	0.07/0.91	0.40/0.87	0.33/0.98	0.00/0.88	0.50/0.97	0.00/0.98	-	-	-	0.14-0.82

Table 2: Precision depending on size, rank and evaluation criteria.

Rank / Size	2	3	4	5	6	7	8	9	10	11	12
0-9999	934-9%	45-0%	1928-19%	218-2%	1764-18%	485-5%	1340-13%	699-7%	977-10%	1031-10%	579-6%
10000-19999	682-7%	100-1%	1662-17%	307-3%	1649-16%	447-4%	1947-19%	629-6%	1299-13%	611-6%	667-7%
20000-29999	1169-12%	243-2%	2969-30%	699-7%	2416-24%	666-7%	748-7%	289-3%	360-4%	265-3%	176-2%
30000-39999	2223-22%	787-8%	3082-31%	948-9%	1160-12%	571-6%	489-5%	303-3%	260-3%	71-1%	106-1%
40000-49999	4191-42%	1545-15%	1690-17%	689-7%	636-6%	375-4%	369-4%	189-2%	194-2%	45-0%	77-1%
Total	15288-27%	3830-7%	11449-20%	2879-5%	7650-13%	2557-4%	4906-9%	2117-4%	3097-5%	2025-4%	1608-3%

Table 3: Number and distribution over size of the translation hypotheses generated.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 597–604, Stroudsburg, PA, USA. Association for Computational Linguistics.

Y. Deng and W. Byrne. 2008. Hmm word and phrase alignment for statistical machine translation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(3):494–507.

William A. Gale and Kenneth W. Church. 1991. Identifying word correspondence in parallel texts. In *Proceedings of the workshop on Speech and Natural Language*, HLT '91, pages 152–157, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Caroline Lavecchia, David Langlois, and Kamel Smaïli. 2008. Phrase-Based Machine Translation based on Simulated Annealing. In European Language Resources Association (ELRA), editor, *Sixth international conference on Language Resources and Evaluation - LREC 2008*, Marrakech, Maroc.

I. Dan Melamed. 1995. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In *In Proceedings of the Third Workshop on Very Large Corpora*, pages 184–198.

I. Dan Melamed. 1997. Automatic discovery of non-compositional compounds in parallel data. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*.

I. Dan Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26:221–249.

Robert C. Moore. 2001. Towards a simple and accurate statistical approach to learning translation relationships among words. In *Proceedings of the workshop on Data-driven methods in machine translation - Volume 14*, DMMT '01, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

H.T. Ng, B. Wang, and Y.S. Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 455–462. Association for Computational Linguistics.

M. Sahlgren and J. Karlgren. 2005. Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Nat. Lang. Eng.*, 11(3):327–341, September.

O. Streiter, M. Stuflesser, and I. Ties. 2004. Cle, an aligned tri-lingual ladin-italian-german corpus. corpus design and interface. *First Steps in Language Documentation for Minority Languages*, page 84.

Keita Tsuji and Kyo Kageura. 2004. Extracting low-frequency translation pairs from japanese-english bilingual corpora. In Sophia Ananadiou and Pierre Zweigenbaum, editors, *COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology*, pages 23–30, Geneva, Switzerland, August 29. COLING.

Dan Tufis. 2002. A cheap and fast way to build useful translation lexicons. In *in Proceedings of the 19th International Conference on Computational Linguistics, COLING2002*, pages 1030–1036.

Dominic Widdows, Beate Dorow, and Chiu ki Chan. 2002. Using parallel corpora to enrich multilingual lexical resources. In *In Third International Conference on Language Resources and Evaluation*, pages 240–245.

Dekai Wu and Xuanyin Xia. 1994. Learning an english-chinese lexicon from a parallel corpus. In *In Proceedings of the First Conference of the Association for Machine Translation in the Americas*, pages 206–213.

Ying Zhang and Stephan Vogel. 2005. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and large corpora. In *In Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT-05*, pages 30–31.