Deciphering Foreign Language by Combining Language Models and Context Vectors

Malte Nuhn and **Arne Mauser*** and **Hermann Ney**
Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Germany
<surname>@cs.rwth-aachen.de

Abstract

In this paper we show how to train statistical machine translation systems on real-life tasks using only non-parallel monolingual data from two languages. We present a modification of the method shown in (Ravi and Knight, 2011) that is scalable to vocabulary sizes of several thousand words. On the task shown in (Ravi and Knight, 2011) we obtain better results with only 5% of the computational effort when running our method with an n-gram language model. The efficiency improvement of our method allows us to run experiments with vocabulary sizes of around 5,000 words, such as a non-parallel version of the VERBMOBIL corpus. We also report results using data from the monolingual French and English GIGAWORD corpora.

1 Introduction

It has long been a vision of science fiction writers and scientists to be able to universally communicate in all languages. In these visions, even previously unknown languages can be learned automatically from analyzing foreign language input.

In this work, we attempt to learn statistical translation models from only monolingual data in the source and target language. The reasoning behind this idea is that the elements of languages share statistical similarities that can be automatically identified and matched with other languages.

This work is a big step towards large-scale and large-vocabulary unsupervised training of statistical translation models. Previous approaches have faced constraints in vocabulary or data size. We show how to scale unsupervised training to real-life translation tasks and how large-scale experiments can be done. Monolingual data is more readily available, if not abundant compared to true parallel or even just translated data. Learning from only monolingual data in real-life translation tasks could improve especially low resource language pairs where few or no parallel texts are available.

In addition to that, this approach offers the opportunity to decipher new or unknown languages and derive translations based solely on the available monolingual data. While we do tackle the full unsupervised learning task for MT, we make some very basic assumptions about the languages we are dealing with:

1. We have large amounts of data available in source and target language. This is not a very strong assumption as books and text on the internet are readily available for almost all languages.

2. We can divide the given text in tokens and sentence-like units. This implies that we know enough about the language to tokenize and sentence-split a given text. Again, for the vast majority of languages, this is not a strong restriction.

3. The writing system is one-dimensional left-to-right. It has been shown (Lin and Knight, 2006) that the writing direction can be determined separately and therefore this assumption does not pose a real restriction.

Previous approaches to unsupervised training for SMT prove feasible only for vocabulary sizes up to around 500 words (Ravi and Knight, 2011) and data

*Author now at Google Inc., amauser@google.com.

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 156–164,
Jeju, Republic of Korea, 8-14 July 2012.

sets of roughly 15,000 sentences containing only about 4 tokens per sentence on average. Real data as it occurs in texts such as web pages or news texts does not meet any of these characteristics.

In this work, we will develop, describe, and evaluate methods for large vocabulary unsupervised learning of machine translation models suitable for real-world tasks. The remainder of this paper is structured as follows: In Section 2, we will review the related work and describe how our approach extends existing work. Section 3 describes the model and training criterion used in this work. The implementation and the training of this model is then described in Section 5 and experimentally evaluated in Section 6.

2 Related Work

Unsupervised training of statistical translations systems without parallel data and related problems have been addressed before. In this section, we will review previous approaches and highlight similarities and differences to our work. Several steps have been made in this area, such as (Knight and Yamada, 1999), (Ravi and Knight, 2008), or (Snyder et al., 2010), to name just a few. The main difference of our work is, that it allows for much larger vocabulary sizes and more data to be used than previous work while at the same time not being dependent on seed lexica and/or any other knowledge of the languages.

Close to the methods described in this work, Ravi and Knight (2011) treat training and translation without parallel data as a deciphering problem. Their best performing approach uses an EM-Algorithm to train a generative word based translation model. They perform experiments on a Spanish/English task with vocabulary sizes of about 500 words and achieve a performance of around 20 BLEU compared to 70 BLEU obtained by a system that was trained on parallel data. Our work uses the same training criterion and is based on the same generative story. However, we use a new training procedure whose critical parts have constant time and memory complexity with respect to the vocabulary size so that our methods can scale to much larger vocabulary sizes while also being faster.

In a different approach, Koehn and Knight (2002) induce a bilingual lexicon from only non-parallel data. To achieve this they use a seed lexicon which they systematically extend by using orthographic as well as distributional features such as context, and frequency. They perform their experiments on non-parallel German-English news texts, and test their mappings against a bilingual lexicon. We use a greedy method similar to (Koehn and Knight, 2002) for extending a given lexicon, and we implicitly also use the frequency as a feature. However, we perform fully unsupervised training and do not start with a seed lexicon or use linguistic features.

Similarly, Haghighi et al. (2008) induce a one-to-one translation lexicon only from non-parallel monolingual data. Also starting with a seed lexicon, they use a generative model based on canonical correlation analysis to systematically extend the lexicon using context as well as spelling features. They evaluate their method on a variety of tasks, ranging from inherently parallel data (EUROPARL) to unrelated corpora ($100k$ sentences of the GIGAWORD corpus). They report F-measure scores of the induced entries between 30 to 70. As mentioned above, our work neither uses a seed lexicon nor orthographic features.

3 Translation Model

In this section, we describe the statistical training criterion and the translation model that is trained using monolingual data. In addition to the mathematical formulation of the model we describe approximations used.

Throughout this work, we denote the source language words as f and target language words as e. The source vocabulary is V_f and we write the size of this vocabulary as $|V_f|$. The same notation holds for the target vocabulary with V_e and $|V_e|$.

As training criterion for the translation model's parameters θ, Ravi and Knight (2011) suggest

$$\arg\max_{\theta} \left\{ \prod_f \sum_e P(e) \cdot p_\theta(f|e) \right\} \qquad (1)$$

We would like to obtain θ from Equation 1 using the EM Algorithm (Dempster et al., 1977). This becomes increasingly difficult with more complex translation models. Therefore, we use a simplified

translation model that still contains all basic phenomena of a generic translation process. We formulate the translation process with the same generative story presented in (Ravi and Knight, 2011):

1. Stochastically generate the target sentence according to an n-gram language model.
2. Insert NULL tokens between any two adjacent positions of the target string with uniform probability.
3. For each target token e_i (including NULL) choose a foreign translation f_i (including NULL) with probability $P_\theta(f_i|e_i)$.
4. Locally reorder any two adjacent foreign words f_{i-1}, f_i with probability $P(\text{SWAP}) = 0.1$.
5. Remove the remaining NULL tokens.

In practice, however, it is not feasible to deal with the full parameter table $P_\theta(f_i|e_i)$ which models the lexicon. Instead we only allow translation models where for each *source* word f the number of words e' with $P(f|e') \neq 0$ is below some fixed value. We will refer to this value as the *maximum number of candidates* of the translation model and denote it with N_C. Note that for a given e this does not necessarily restrict the number of entries $P(f'|e) \neq 0$. Also note that with a fixed value of N_C, time and memory complexity of the EM step is $\mathcal{O}(1)$ with respect to $|V_e|$ and $|V_f|$.

In the following we divide the problem of maximizing Equation 1 into two parts:

1. Determining a set of active lexicon entries.
2. Choosing the translation probabilities for the given set of active lexicon entries.

The second task can be achieved by running the EM algorithm on the restricted translation model. We deal with the first task in the following section.

4 Monolingual Context Similarity

As described in Section 3 we need some mechanism to iteratively choose an active set of translation candidates. Based on the assumption that some of the active candidates and their respective probabilities are already correct, we induce new active candidates. In the context of information retrieval, Salton et al. (1975) introduce a document space where each document identified by one or more index terms is represented by a high dimensional vector of term weights. Given two vectors v_1 and v_2 of two documents it is then possible to calculate a similarity coefficient between those given documents (which is usually denoted as $s(v_1, v_2)$). Similar to this we represent source and target words in a high dimensional vector space of target word weights which we call *context vectors* and use a similarity coefficient to find possible translation pairs. We first initialize these context vectors using the following procedure:

1. Using only the monolingual data for the target language, prepare the context vectors v_{e_i} with entries v_{e_i,e_j}:
   (a) Initialize all $v_{e_i,e_j} = 0$
   (b) For each target sentence E:
   For each word e_i in E:
   For each word $e_j \neq e_i$ in E:
   $v_{e_i,e_j} = v_{e_i,e_j} + 1.$
   (c) Normalize each vector v_{e_i} such that $\sum_{e_j}(v_{e_i,e_j})^2 \stackrel{!}{=} 1$ holds.

   Using the notation $\underline{e_i} = \big(e_j : v_{e_i,e_j}, \dots\big)$ these vectors might for example look like

   $$\underline{work} = (early : 0.2, late : 0.1, \dots)$$
   $$\underline{time} = (early : 0.2, late : 0.2, \dots).$$

2. Prepare context vectors v_{f_i,e_j} for the source language using only the monolingual data for the source language and the translation model's current parameter estimate θ:
   (a) Initialize all $v_{f_i,e_j} = 0$
   (b) Let $\tilde{E}_\theta(F)$ denote the most probable translation of the foreign sentence F obtained by using the current estimate θ.
   (c) For each source sentence F:
   For each word f_i in F:
   For each word $e_j \neq E_\theta(f_i)$[1] in $E_\theta(F)$:
   $v_{f_i,e_j} = v_{f_i,e_j} + 1$
   (d) Normalize each vector v_{f_i} such that $\sum_{e_j}(v_{f_i,e_j})^2 \stackrel{!}{=} 1$ holds.

[1] denoting that e_j is not the translation of f_i in $E_\theta(F)$

Adapting the notation described above, these vectors might for example look like

$$\underline{Arbeit} = (early : 0.25, late : 0.05, \dots)$$
$$\underline{Zeit} = (early : 0.15, late : 0.25, \dots)$$

Once we have set up the context vectors v_e and v_f, we can retrieve translation candidates for some source word f by finding those words e' that maximize the similarity coefficient $s(v_{e'}, v_f)$, as well as candidates for a given target word e by finding those words f' that maximize $s(v_e, v_{f'})$. In our implementation we use the Euclidean distance

$$d(v_e, v_f) = ||v_e - v_f||_2. \quad (2)$$

as *distance* measure.[2] The normalization of context vectors described above is motivated by the fact that the context vectors should be invariant with respect to the absolute number of occurrences of words.[3]

Instead of just finding the best candidates for a given word, we are interested in an assignment that involves all source and target words, minimizing the sum of distances between the assigned words. In case of a one-to-one mapping the problem of assigning translation candidates such that the sum of distances is minimal can be solved optimally in polynomial time using the *hungarian algorithm* (Kuhn, 1955). In our case we are dealing with a many-to-many assignment that needs to satisfy the *maximum number of candidates* constraints. For this, we solve the problem in a greedy fashion by simply choosing the best pairs (e, f) first. As soon as a target word e or source word f has reached the limit of maximum candidates, we skip all further candidates for that word e (or f respectively). This step involves calculating and sorting all $|V_e| \cdot |V_f|$ distances which can be done in time $\mathcal{O}(V^2 \cdot \log(V))$, with $V = \max(|V_e|, |V_f|)$. A simplified example of this procedure is depicted in Figure 1. The example already shows that the assignment obtained by this algorithm is in general not optimal.

[2] We then obtain pairs (e, f) that *minimize* d.

[3] This gives the same similarity ordering as using unnormalized vectors with the *cosine similarity measure* $\frac{v_e \cdot v_f}{||v_e||_2 \cdot ||v_f||_2}$ which can be interpreted as measuring the cosine of the angle between the vectors, see (Manning et al., 2008). Still it is noteworthy that this procedure is not equivalent to the *tf-IDF* context vectors described in (Salton et al., 1975).

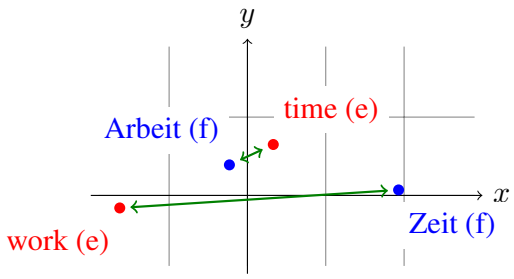

Figure 1: Hypothetical example for a greedy one-to-one assignment of translation candidates. The optimal assignment would contain (time,Zeit) and (work,Arbeit).

5 Training Algorithm and Implementation

Given the model presented in Section 3 and the methods illustrated in Section 4, we now describe how to train this model.

As described in Section 4, the overall procedure is divided into two alternating steps: After initialization we first perform EM training of the translation model for 20-30 iterations using a 2-gram or 3-gram language model in the target language. With the obtained best translations we induce new translation candidates using context similarity. This procedure is depicted in Figure 2.

5.1 Initialization

Let N_C be the maximum number of candidates per source word we allow, V_e and V_f be the target/source vocabulary and $r(e)$ and $r(f)$ the frequency rank of a source/target word. Each word $f \in V_f$ with frequency rank $r(f)$ is assigned to all words $e \in V_e$ with frequency rank

$$r(e) \in [\,\mathrm{start}(f)\,,\,\mathrm{end}(f)\,] \quad (3)$$

where

$$\mathrm{start}(f) = \max(0\,,\,\min\left(|V_e| - N_c\,,\,\left\lfloor \frac{|V_e|}{|V_f|} \cdot r(f) - \frac{N_c}{2} \right\rfloor\right)) \quad (4)$$

$$\mathrm{end}(f) = \min\left(start(f) + N_c, |V_e|\right). \quad (5)$$

This defines a diagonal beam[4] when visualizing the lexicon entries in a matrix where both source and target words are sorted by their frequency rank. However, note that the result of sorting by frequency

[4] The diagonal has some artifacts for the highest and lowest frequency ranks. See, for example, left side of Figure 2.

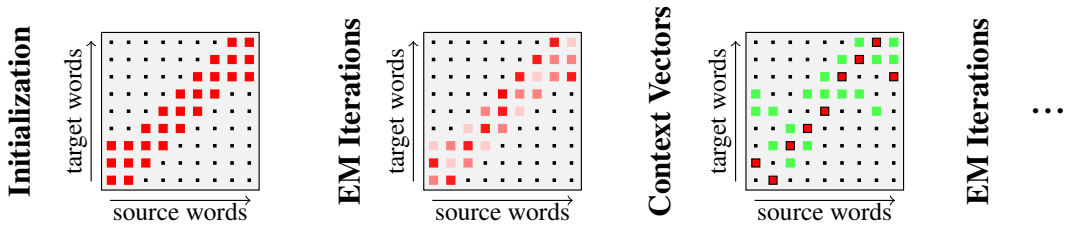

Figure 2: Visualization of the training procedure. The big rectangles represent word lexica in different stages of the training procedure. The small rectangles represent word pairs (e, f) for which e is a translation candidate of f, while dots represent word pairs (e, f) for which this is not the case. Source and target words are sorted by frequency so that the most frequent source words appear on the very left, and the most frequent target words appear at the very bottom.

and thus the frequency ranks are not unique when there are words with the same frequency. In this case, we initially obtain some not further specified frequency ordering, which is then kept throughout the procedure.

This initialization proves useful as we show by taking an IBM1 lexicon $P(f|e)$ extracted on the parallel VERBMOBIL corpus (Wahlster, 2000): For each word e we calculate the weighted rank difference

$$\Delta r_{\text{avg}}(e) = \sum_f P(f|e) \cdot |(r(e) - r(f)| \quad (6)$$

and count how many of those weighted rank differences are smaller than a given value $\frac{N_C}{2}$. Here we see that for about 1% of the words the weighted rank difference lies within $N_C = 50$, and even about 3% for $N_C = 150$ respectively. This shows that the initialization provides a first solid guess of possible translations.

5.2 EM Algorithm

The generative story described in Section 3 is implemented as a cascade of a permutation, insertion, lexicon, deletion and language model finite state transducers using OpenFST (Allauzen et al., 2007). Our FST representation of the LM makes use of failure transitions as described in (Allauzen et al., 2003). We use the forward-backward algorithm on the composed transducers to efficiently train the lexicon model using the EM algorithm.

5.3 Context Vector Step

Given the trained parameters θ from the previous run of the EM algorithm we set the context vectors v_e and v_f up as described in Section 4. We then calculate and sort all $|V_e| \cdot |V_f|$ distances which proves feasible in a few CPU hours even for vocabulary sizes of more than 50,000 words. This is achieved with the GNU SORT tool, which uses external sorting for sorting large amounts of data.

To set up the new lexicon we keep the $\lfloor \frac{N_C}{2} \rfloor$ best translations for each source word with respect to $P(e|f)$, which we obtained in the previous EM run. Experiments showed that it is helpful to also limit the number of candidates per target words. We therefore prune the resulting lexicon using $P(f|e)$ to a maximum of $\lfloor \frac{N'_C}{2} \rfloor$ candidates per target word afterwards. Then we fill the lexicon with new candidates using the previously sorted list of candidate pairs such that the final lexicon has at most N_C candidates per source word and at most N'_C candidates per target word. We set N'_C to some value $N'_C > N_C$. All experiments in this work were run with $N'_C = 300$. Values of $N'_C \approx N_C$ seem to produce poorer results. Not limiting the number of candidates per target word at all also typically results in weaker performance. After the lexicon is filled with candidates, we initialize the probabilities to be uniform. With this new lexicon the process is iterated starting with the EM training.

6 Experimental Evaluation

We evaluate our method on three different corpora.

At first we apply our method to non-parallel Spanish/English data that is based on the OPUS corpus (Tiedemann, 2009) and that was also used in (Ravi and Knight, 2011). We show that our method performs better by 1.6 BLEU than the best performing method described in (Ravi and Knight, 2011) while

Name	Lang.	Sent.	Words	Voc.
OPUS	Spanish	13,181	39,185	562
	English	19,770	61,835	411
VERBMOBIL	German	27,861	282,831	5,964
	English	27,862	294,902	3,723
GIGAWORD	French	100,000	1,725,993	68,259
	English	100,000	1,788,025	64,621

Table 1: Statistics of the corpora used in this paper.

being approximately 15 to 20 times faster than their n-gram based approach.

After that we apply our method to a non-parallel version of the German/English VERBMOBIL corpus, which has a vocabulary size of 6,000 words on the German side, and 3,500 words on the target side and which thereby is approximately one order of magnitude larger than the previous OPUS experiment.

We finally run our system on a subset of the non-parallel French/English GIGAWORD corpus, which has a vocabulary size of 60,000 words for both French and English. We show first interesting results on such a big task.

In case of the OPUS and VERBMOBIL corpus, we evaluate the results using BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) to reference translations. We report all scores in percent. For BLEU higher values are better, for TER lower values are better. We also compare the results on these corpora to a system trained on parallel data.

In case of the GIGAWORD corpus we show lexicon entries obtained during training.

6.1 OPUS Subtitle Corpus

6.1.1 Experimental Setup

We apply our method to the corpus described in Table 6. This exact corpus was also used in (Ravi and Knight, 2011). The best performing methods in (Ravi and Knight, 2011) use the full 411×579 lexicon model and apply standard EM training. Using a 2-gram LM they obtain 15.3 BLEU and with a whole segment LM, they achieve 19.3 BLEU. In comparison to this baseline we run our algorithm with $N_C = 50$ candidates per source word for both, a 2-gram and a 3-gram LM. We use 30 EM iterations between each context vector step. For both cases we run 7 EM+Context cycles.

6.1.2 Results

Figure 3 and Figure 4 show the evolution of BLEU and TER scores for applying our method using a 2-gram and a 3-gram LM.

In case of the 2-gram LM (Figure 3) the translation quality increases until it reaches a plateau after 5 EM+Context cycles. In case of the 3-gram LM (Figure 4) the statement only holds with respect to TER. It is notable that during the first iterations TER only improves very little until a large chunk of the language unravels after the third iteration. This behavior may be caused by the fact that the corpus only provides a relatively small amount of context information for each word, since sentence lengths are 3-4 words on average.

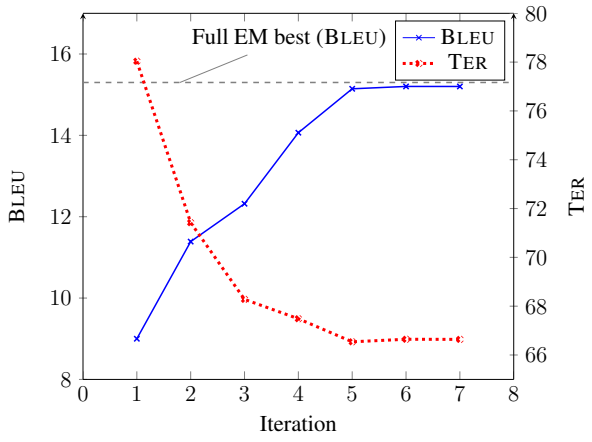

Figure 3: Results on the OPUS corpus with a 2-gram LM, $N_C = 50$, and 30 EM iterations between each context vector step. The dashed line shows the best result using a 2-gram LM in (Ravi and Knight, 2011).

Table 2 summarizes these results and compares them with (Ravi and Knight, 2011). Our 3-gram based method performs by 1.6 BLEU better than their best system which is a statistically significant improvement at 95% confidence level. Furthermore, Table 2 compares the CPU time needed for training. Our 3-gram based method is 15-20 times faster than running the EM based training procedure presented in (Ravi and Knight, 2011) with a 3-gram LM[5].

[5](Ravi and Knight, 2011) only report results using a 2-gram LM and a whole-segment LM.

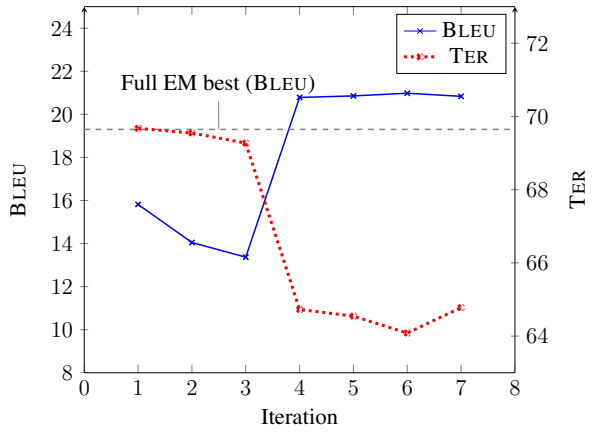

Figure 4: Results on the OPUS corpus with a 3-gram LM, $N_C = 50$, and 30 EM iterations between each context vector step. The dashed line shows the best result using a whole-segment LM in (Ravi and Knight, 2011)

Method	CPU	BLEU	TER
EM, 2-gram LM 411 cand. p. source word (Ravi and Knight, 2011)	≈850h[6]	15.3	–
EM, Whole-segment LM 411 cand. p. source word (Ravi and Knight, 2011)	–[7]	19.3	–
EM+Context, 2-gram LM 50 cand. p. source word (this work)	**50h**[8]	**15.2**	**66.6**
EM+Context, 3-gram LM 50 cand. p. source word (this work)	**200h**[8]	**20.9**	**64.5**

Table 2: Results obtained on the OPUS corpus.

To summarize: Our method is significantly faster than n-gram LM based approaches and obtains better results than any previously published method.

[6]Estimated by running full EM using the 2-gram LM using our implementation for 90 Iterations yielding 15.2 BLEU.

[7]≈4,000h when running full EM using a 3-gram LM, using our implementation. Estimated by running only the first iteration and by assuming that the final result will be obtained after 90 iterations. However, (Ravi and Knight, 2011) report results using a whole segment LM, assigning $P(e) > 0$ only to sequences seen in training. This seems to work for the given task but we believe that it can not be a general replacement for higher order n-gram LMs.

[8]Estimated by running our method for 5×30 iterations.

6.2 VERBMOBIL Corpus

6.2.1 Experimental Setup

The VERBMOBIL corpus is a German/English corpus dealing with short sentences for making appointments. We prepared a non-parallel subset of the original VERBMOBIL (Wahlster, 2000) by splitting the corpus into two parts and then selecting only the German side from the first half, and the English side from the second half such that the target side is not the translation of the source side. The source and target vocabularies of the resulting non-parallel corpus are both more than 9 times bigger compared to the OPUS vocabularies. Also the total amount of word tokens is more than 5 times larger compared to the OPUS corpus. Table 6 shows the statistics of this corpus. We run our method for 5 EM+Context cycles (30 EM iterations each) using a 2-gram LM. After that we run another five EM+Context cycles using a 3-gram LM.

6.2.2 Results

Our results on the VERBMOBIL corpus are summarized in Table 3. Even on this more complex task our method achieves encouraging results: The translation quality increases from iteration to iteration until the algorithm finally reaches 11.7 BLEU using only the 2-gram LM. Running further five cycles using a 3-gram LM achieves a final performance of 15.5 BLEU. Och (2002) reports results of 48.2 BLEU for a single-word based translation system and 56.1 BLEU using the alignment template approach, both trained on parallel data. However, it should be noted that our experiment only uses 50% of the original VERBMOBIL training data to simulate a truly non-parallel setup.

Method	BLEU	TER
5×30 Iterations EM+Context 50 cand. p. source word, 2-gram LM	**11.7**	**67.4**
+ 5×30 Iterations EM+Context 50 cand. p. source word, 3-gram LM	**15.5**	**63.2**

Table 3: Results obtained on the VERBMOBIL corpus.

Iter.	e	$p(f_1\|e)$	f_1	$p(f_2\|e)$	f_2	$p(f_3\|e)$	f_3	$p(f_4\|e)$	f_4	$p(f_5\|e)$	f_5
1.	**the**	0.43	*la*	0.31	*l'*	0.11	une	0.04	*le*	0.04	*les*
2.	**several**	0.57	*plusieurs*	0.21	les	0.09	des	0.03	*nombreuses*	0.02	deux
3.	**where**	0.63	*où*	0.17	mais	0.06	indique	0.04	précise	0.02	appelle
4.	**see**	0.49	éviter	0.09	effet	0.09	*voir*	0.05	envisager	0.04	dire
5.	**January**	0.25	octobre	0.22	mars	0.09	juillet	0.07	août	0.07	*janvier*
—	**Germany**	0.24	Italie	0.12	Espagne	0.06	Japon	0.05	retour	0.05	Suisse

Table 4: Lexicon entries obtained by running our method on the non-parallel GIGAWORD corpus. The first column shows in which iteration the algorithm found the first correct translations f (compared to a parallely trained lexicon) among the top 5 candidates

6.3 GIGAWORD

6.3.1 Experimental Setup

This setup is based on a subset of the monolingual GIGAWORD corpus. We selected 100,000 French sentences from the news agency *AFP* and 100,000 sentences from the news agency *Xinhua*. To have a more reliable set of training instances, we selected only sentences with more than 7 tokens. Note that these corpora form true non-parallel data which, besides the length filtering, were not specifically preselected or pre-processed. More details on these non-parallel corpora are summarized in Table 6. The vocabularies have a size of approximately 60,000 words which is more than 100 times larger than the vocabularies of the OPUS corpus. Also it incorporates more than 25 times as many tokens as the OPUS corpus.

After initialization, we run our method with $N_C = 150$ candidates per source word for 20 EM iterations using a 2-gram LM. After the first context vector step with $N_C = 50$ we run another 4×20 iterations with $N_C = 50$ with a 2-gram LM.

6.3.2 Results

Table 4 shows example lexicon entries we obtained. Note that we obtained these results by using purely non-parallel data, and that we neither used a seed lexicon, nor orthographic features to assign e.g. numbers or proper names: All results are obtained using 2-gram statistics and the context of words only. We find the results encouraging and think that they show the potential of large-scale unsupervised techniques for MT in the future.

7 Conclusion

We presented a method for learning statistical machine translation models from non-parallel data. The key to our method lies in limiting the translation model to a limited set of translation candidates and then using the EM algorithm to learn the probabilities. Based on the translations obtained with this model we obtain new translation candidates using a context vector approach. This method increased the training speed by a factor of 10-20 compared to methods known in literature and also resulted in a 1.6 BLEU point increase compared to previous approaches. Due to this efficiency improvement we were able to tackle larger tasks, such as a non-parallel version of the VERBMOBIL corpus having a nearly 10 times larger vocabulary. We also had a look at first results of our method on an even larger Task, incorporating a vocabulary of 60,000 words. We have shown that, using a limited set of translation candidates, we can significantly reduce the computational complexity of the learning task. This work serves as a big step towards large-scale unsupervised training for statistical machine translation systems.

Acknowledgements

This work was realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The authors would like to thank Sujith Ravi and Kevin Knight for providing us with the OPUS subtitle corpus and David Rybach for kindly sharing his knowledge about the OpenFST library.

References

Cyril Allauzen, Mehryar Mohri, and Brian Roark. 2003. Generalized algorithms for constructing statistical language models. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 40–47. Association for Computational Linguistics.

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. Openfst: A general and efficient weighted finite-state transducer library. In Jan Holub and Jan Zdárek, editors, *CIAA*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. Springer.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39.

Aria Haghighi, Percy Liang, T Berg-Kirkpatrick, and Dan Klein. 2008. Learning Bilingual Lexicons from Monolingual Corpora. In *Proceedings of ACL08 HLT*, pages 771–779. Association for Computational Linguistics.

Kevin Knight and Kenji Yamada. 1999. A computational approach to deciphering unknown scripts. In *ACL Workshop on Unsupervised Learning in Natural Language Processing*, number 1, pages 37–44. Citeseer.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL02 workshop on Unsupervised lexical acquisition*, number July, pages 9–16. Association for Computational Linguistics.

Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97.

Shou-de Lin and Kevin Knight. 2006. Discovering the linear writing order of a two-dimensional ancient hieroglyphic script. *Artificial Intelligence*, 170:409–421, April.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schuetze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July.

Franz J. Och. 2002. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, October.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2008. Attacking decipherment problems optimally with low-order n-gram models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 812–819, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 12–21, Portland, Oregon, USA, June. Association for Computational Linguistics.

Gerard M. Salton, Andrew K. C. Wong, and Chang S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA, August.

Benjamin Snyder, Regina Barzilay, and Kevin Knight. 2010. A statistical model for lost language decipherment. In *48th Annual Meeting of the Association for Computational Linguistics*, number July, pages 1048–1057.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Wolfgang Wahlster, editor. 2000. *Verbmobil: Foundations of speech-to-speech translations*. Springer-Verlag, Berlin.