Unsupervised Training for Large Vocabulary Translation Using Sparse Lexicon and Word Classes

Yunsu Kim, Julian Schamper and **Hermann Ney**
Human Language Technology and Pattern Recognition Group
RWTH Aachen University
{surname}@cs.rwth-aachen.de

Abstract

We address for the first time unsupervised training for a translation task with hundreds of thousands of vocabulary words. We scale up the expectation-maximization (EM) algorithm to learn a large translation table without any parallel text or seed lexicon. First, we solve the memory bottleneck and enforce the sparsity with a simple thresholding scheme for the lexicon. Second, we initialize the lexicon training with word classes, which efficiently boosts the performance. Our methods produced promising results on two large-scale unsupervised translation tasks.

1 Introduction

Statistical machine translation (SMT) heavily relies on parallel text to train translation models with supervised learning. Unfortunately, parallel training data is scarce for most language pairs, where an alternative learning formalism is highly in need.

In contrast, there is a virtually unlimited amount of monolingual data available for most languages. Based on this fact, we define a basic *unsupervised learning problem for SMT* as follows; given only a source text of arbitrary length and a target side LM, which is built from a huge target monolingual corpus, we are to learn translation probabilities of all possible source-target word pairs.

We solve this problem using the EM algorithm, updating the translation hypothesis of the source text over the iterations. In a very large vocabulary setup, the algorithm has two fundamental problems: 1) A full lexicon table is too large to keep in memory during the training. 2) A search space for hypotheses grows exponentially with the vocabulary size, where both memory and time requirements for the forward-backward step explode.

For this condition, it is unclear how the lexicon can be efficiently represented and whether the training procedure will work and converge properly. This paper answers these questions by 1) filtering out unlikely lexicon entries according to the training progress and 2) using word classes to learn a stable starting point for the training. For the first time, we eventually enabled the EM algorithm to translate 100k-vocabulary text in an unsupervised way, achieving 54.2% accuracy on EUROPARL Spanish→English task and 32.2% on IWSLT 2014 Romanian→English task.

2 Related Work

Early work on unsupervised sequence learning was mainly for *deterministic decipherment*, a combinatorial problem of matching input-output symbols with 1:1 or homophonic assumption (Knight et al., 2006; Ravi and Knight, 2011a; Nuhn et al., 2013). *Probabilistic decipherment* relaxes this assumption to allow many-to-many mapping, while the vocabulary is usually limited to a few thousand types (Nuhn et al., 2012; Dou and Knight, 2013; Nuhn and Ney, 2014; Dou et al., 2015).

There has been several attempts to improve the scalability of decipherment methods, which are however not applicable to 100k-vocabulary translation scenarios. For EM-based decipherment, Nuhn et al. (2012) and Nuhn and Ney (2014) accelerate hypothesis expansions but do not explicitly solve the memory issue for a large lexicon table. Count-based Bayesian inference (Dou and Knight, 2012; Dou and Knight, 2013; Dou et al., 2015) loses all context information beyond bigrams for the sake of efficiency; it is therefore particularly effective in contextless deterministic ciphers or in inducing an auxiliary lexicon for supervised SMT. Ravi (2013) uses binary hashing to quicken the Bayesian sampling procedure, which

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 650–656,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

yet shows poor performance in large-scale experiments.

Our problem is also related to *unsupervised tagging* with hidden Markov model (HMM). To the best of our knowledge, there is no published work on HMM training for a 100k-size discrete space. HMM taggers are often integrated with sparse priors (Goldwater and Griffiths, 2007; Johnson, 2007), which is not readily possible in a large vocabulary setting due to the memory bottleneck.

Learning a good initialization on a smaller model is inspired by Och and Ney (2003) and Knight et al. (2006). Word classes have been widely used in SMT literature as factors in translation (Koehn and Hoang, 2007; Rishøj and Søgaard, 2011) or smoothing space of model components (Wuebker et al., 2013; Kim et al., 2016).

3 Baseline Framework

Unsupervised learning is yet computationally demanding to solve general translation tasks including reordering or phrase translation. Instead, we take a simpler task which assumes 1:1 monotone alignment between source and target words. This is a good initial test bed for unsupervised translation, where we remove the reordering problem and focus on the lexicon training.

Here is how we set up our unsupervised task: We rearranged the source words of a parallel corpus to be monotonically aligned to the target words and removed multi-aligned or unaligned words, according to the learned word alignments. The corpus was then divided into two parts, using the source text of the first part as an input (f_1^N) and the target text of the second part as LM training data. In the end, we are given only monolingual part of each side which is not sentence-aligned. The statistics of the preprocessed corpora for our experiments are given in Table 1.

Task		Source	Target
		(Input)	(LM)
EuTrans	Run. Words	85k	4.2M
es-en	Vocab.	677	505
EuroParl	Run. Words	2.7M	42.9M
es-en	Vocab.	32k	**96k**
IWSLT	Run. Words	2.8M	13.7M
ro-en	Vocab.	**99k**	**114k**

Table 1: Corpus statistics.

To evaluate a translation output $\hat{e}_1^N$, we use token-level accuracy (Acc.):

$$\text{Acc.} = \frac{\sum\limits_{n=1}^{N} [\hat{e}_n = r_n]}{N} \qquad (1)$$

where r_1^N is the reference output which is the target text of the first division of the corpus. It aggregates all true/false decisions on each word position, comparing the hypothesis with the reference. This can be regarded as the inverse of word error rate (WER) without insertions and deletions. It is simple to understand and nicely fits to our reordering-free task.

In the following, we describe a baseline method to solve this task. For more details, we refer the reader to Schamper (2015).

3.1 Model

We adopt a noisy-channel approach to define a joint probability of f_1^N and e_1^N as follows:

$$p(e_1^N, f_1^N) = \prod_{n=1}^{N} p(e_n|e_{n-m+1}^{n-1})\, p(f_n|e_n) \qquad (2)$$

which is composed of a pre-trained m-gram target LM and a word-to-word translation model. The translation model is parametrized by a full table over the entire source and target vocabularies:

$$p(f|e) = \theta_{f|e} \qquad (3)$$

with normalization constraints $\forall_e \sum_f \theta_{f|e} = 1$. Having this model, the best hypothesis $\hat{e}_1^N$ is obtained by the Viterbi decoding.

3.2 Training

To learn the lexicon parameters $\{\theta\}$, we use maximum likelihood estimation. Since a reference translation is not given, we treat e_1^N as a latent variable and use the EM algorithm (Dempster et al., 1977) to train the lexicon model. The update equation for each maximization step (M-step) of the algorithm is:

$$\hat{\theta}_{f|e} = \frac{\sum\limits_{n:\, f_n = f} p_n(e|f_1^N)}{\sum\limits_{f'} \sum\limits_{n':\, f_{n'} = f'} p_{n'}(e|f_1^N)} \qquad (4)$$

with $p_n(e|f_1^N) = \sum_{e_1^N : e_n = e} p(e_1^N|f_1^N)$. This quantity is computed by the forward-backward algorithm in the expectation step (E-step).

4 Sparse Lexicon

Loading a full table lexicon (Equation 3) is infeasible for very large vocabularies. As only a few f's may be eligible translations of a target word e, we propose a new lexicon model which keeps only those entries with a probability of at least τ:

$$\mathcal{F}(e) = \{f \mid \hat{\theta}_{f|e} \geq \tau\} \tag{5}$$

$$p_{\text{sp}}(f|e) = \begin{cases} \dfrac{\hat{\theta}_{f|e}}{\sum\limits_{f' \in \mathcal{F}(e)} \hat{\theta}_{f'|e}} & \text{if } f \in \mathcal{F}(e) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

We call this model *sparse* lexicon, because only a small percentage of full lexicon is *active*, i.e. has nonzero probability.

The thresholding by τ allows flexibility in the number of active entries over different target words. If e has little translation ambiguity, i.e. probability mass of $\theta_{f|e}$ is concentrated at only a few f's, $p_{\text{sp}}(f|e)$ occupies smaller memory than other more ambiguous target words. For each M-step update, it reduces its size on the fly as we learn sparser E-step posteriors.

However, the sparse lexicon might exclude potentially important entries in early training iterations, when the posterior estimation is still not reliable. Once an entry has zero probability, it can never be recovered by the EM algorithm afterwards. A naive workaround is to adjust the threshold during the training, but it does not actually help for the performance in our internal experiments.

To give a chance to zero-probability translations throughout the training, we smooth the sparse lexicon with a backoff model $p_{\text{bo}}(f)$:

$$p(f|e) = \lambda \cdot p_{\text{sp}}(f|e) + (1 - \lambda) \cdot p_{\text{bo}}(f) \tag{7}$$

where λ is the interpolation parameter. As a backoff model, we use uniform distribution, unigram of source words, or Kneser-Ney lower order model (Kneser and Ney, 1995; Foster et al., 2006).

In Table 2, we illustrate the effect of the sparse lexicon with EUTRANS Spanish→English task (Amengual et al., 1996), comparing to the existing EM decipherment approach (full lexicon). By setting the threshold small enough ($\tau = 0.001$), the sparse lexicon surpasses the performance of the full lexicon, while the number of active entries, for which memory is actually allocated, is greatly reduced. For the backoff, the uniform model shows the best performance, which requires no additional memory. The time complexity is not increased by using the new lexicon.

Lexicon	τ	p_{bo}	Acc. [%]	Active Entries [%]
Full	-	-	70.2	100
Sparse	0.01	Uniform	64.0	1.1
	0.005		69.0	2.7
	0.001		**71.8**	**6.3**
	0.001	Unigram	71.3	6.2
		Kneser-Ney	71.4	6.4

Table 2: Sparse lexicon with different threshold values and backoff models ($\lambda = 0.99$). Initialized with uniform distributions and trained for 50 iterations with a bigram LM. No pruning is applied.

We also study the mutual effect of τ and λ (Figure 1). For a larger τ (circles), where many entries are cut out from the lexicon, the best-performing λ gets smaller ($\lambda = 0.1$). In contrast, when we lower the threshold enough (squares), the performance is more robust to the change of λ, while a higher weight on the trained lexicon ($\lambda = 0.7$) works best. This means that, the higher the threshold is set, the more information we lose and the backoff model plays a bigger role, and vice versa.

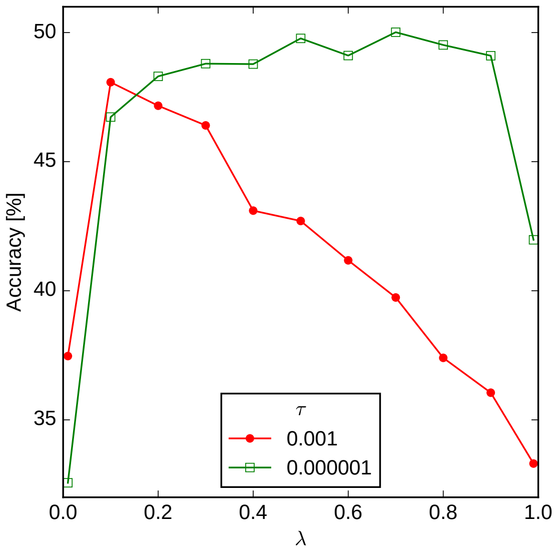

Figure 1: Relation between sparse lexicon parameters (EUROPARL Spanish→English task).

The idea of filtering and smoothing parameters in the EM training is relevant to Deligne and Bimbot (1995) and Marcu and Wong (2002). They leave out a fixed set of parameters for the whole training process, while we update trainable parameters for every iteration. Nuhn and Ney (2014) also perform an analogous smoothing but without filtering, only to moderate the lattice pruning. Note that our work is distinct from the conventional pruning of translation tables in supervised SMT which is applied after the entire training.

5 Initialization Using Word Classes

Apart from the memory problem, it is inevitable to apply pruning in the forward-backward algorithm for runtime efficiency. The pruning in early iterations, however, may drop chances to find a better optimum in later stage of training. One might suggest to prune only for later iterations, but for large vocabularies, a single non-pruned E-step can blow up the total training time.

We rather stabilize the training by a proper initialization of the parameters, so that the training is less worsened by early pruning. We learn an initial lexicon on automatically clustered word classes (Martin et al., 1998), following these steps:

1. Estimate word-class mappings on both sides $(\mathcal{C}_{\text{src}}, \mathcal{C}_{\text{tgt}})$
2. Replace each word in the corpus with its class
$$\begin{aligned} f &\mapsto \mathcal{C}_{\text{src}}(f) \\ e &\mapsto \mathcal{C}_{\text{tgt}}(e) \end{aligned}$$
3. Train a class-to-class full lexicon with a target class LM
4. Convert 3 to an unnormalized word lexicon by mapping each class back to its member words
$$\forall(f, e) \quad q(f|e) := p(\mathcal{C}_{\text{src}}(f)|\,\mathcal{C}_{\text{tgt}}(e))$$
5. Apply the thresholding on 4 and renormalize (Equation 6)

where all f's in an implausible source class are left out together from the lexicon. The resulting distribution $p_{\text{sp}}(f|e)$ is identical for all e's in the same target class.

Word classes group words by syntactic or semantic similarity (Brown et al., 1992), which serve as a reasonable approximation of the original word vocabulary. They are especially suitable for large vocabulary data, because one can arbitrarily choose the number of classes to be very small; learning a class lexicon can thus be much more efficient than learning a word lexicon.

Initialization			Acc. [%]
Uniform			63.7
	#Classes	Class LM	
Word Classes	25	2-gram	67.4
	50	2-gram	69.1
	100	2-gram	72.1
	50	3-gram	76.0
	50	4-gram	**76.2**

Table 3: Sparse lexicon with word class initialization ($\tau = 0.001$, $\lambda = 0.99$, uniform backoff). Pruning is applied with histogram size 10.

Table 3 shows that translation quality is consistently enhanced by the word class initialization, which compensates the performance loss caused by harsh pruning. With a larger number of classes, we have a more precise pre-estimate of the sparse lexicon and thus have more performance gain. Due to the small vocabulary size, we are comfortable to use higher order class LM, which yields even better accuracy, outperforming the non-pruned results of Table 2. The memory and time requirements are only marginally affected by the class lexicon training.

Empirically, we find that the word classes do not really distinguish different conjugations of verbs or nouns. Even if we increase the number of classes, they tend to subdivide the vocabulary more based on semantics, keeping morphological variations of a word in the same class. From this fact, we argue that the word class initialization can be generally useful for language pairs with different roots. We also emphasize that word classes are estimated without any model training or language-specific annotations. This is a clear advantage for unknown/historic languages, where the unsupervised translation is indeed in need.

6 Large Vocabulary Experiments

We applied two proposed techniques to EUROPARL Spanish→English corpus (Koehn, 2005) and IWSLT 2014 Romanian→English TED talk corpus (Cettolo et al., 2012). In the EUROPARL data, we left out long sentences with more than 25 words and sentences with singletons. For the IWSLT data, we extended the LM training part with news commentary corpus from WMT 2016 shared tasks.

We learned the initial lexicons on 100 classes for both sides, using 4-gram class LMs with 50 EM iterations. The sparse lexicons were trained with trigram LMs for 100 iterations ($\tau = 10^{-6}$, $\lambda = 0.15$). For further speedup, we applied per-position pruning with histogram size 50 and the preselection method of Nuhn and Ney (2014) with lexical beam size 5 and LM beam size 50. All our experiments were carried out with the UNRAVEL toolkit (Nuhn et al., 2015).

Table 4 summarizes the results. The supervised learning scores were obtained by decoding with an optimal lexicon estimated from the input text and its reference. Our methods achieve significantly high accuracy with only less than 0.1% of memory for the full lexicon. Note that using conventional decipherment methods is impossible to conduct these scales of experiments.

	Acc. [%]		
Task	Supervised	Unsupervised	Lex. Size [%]
es-en	77.5	**54.2**	**0.06**
ro-en	72.3	**32.2**	**0.03**

Table 4: Large vocabulary translation results.

7 Conclusion and Future Work

This paper has shown the first promising results on 100k-vocabulary translation with no bilingual data. To facilitate this, we proposed the sparse lexicon, which effectively emphasizes the multinomial sparsity and minimizes its memory usage throughout the training. In addition, we described how to learn an initial lexicon on word class vocabulary for a robust training. Note that one can optimize the performance to a given computing environment by tuning the lexicon threshold, the number of classes, and the class LM order.

Nonetheless, we still observe a substantial difference in performance between supervised and unsupervised learning for large vocabulary translation. We will exploit more powerful LMs and more input text to see if this gap can be closed. This may require a strong approximation with respect to numerous LM states along with an online algorithm.

As a long term goal, we plan to relax constraints on word alignments to make our framework usable for more realistic translation scenarios. The first step would be modeling local reorderings such as insertions, deletions, and/or local swaps (Ravi and Knight, 2011b; Nuhn et al., 2012). Note that the idea of thresholding in the sparse lexicon is also applicable to any normalized model components. When the reordering model is lexicalized, the word class initialization may also be helpful for a stable training.

Acknowledgments

This work was supported by the Nuance Foundation and also received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n^{o} 645452 (QT21).

References

Juan-Carlos Amengual, José-Miguel Benedí, Asunción Castaño, Andrés Marzal, Federico Prat, Enrique Vidal, Juan Miguel Vilar, Cristina Delogu, Andrea Di Carlo, Hermann Ney, and Stephan Vogel. 1996. Definition of a machine translation task and generation of corpora. Technical report, EUTRANS (IT-LTR-OS-20268).

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, December.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT 2012)*, pages 261–268, Trento, Italy, May.

Sabine Deligne and Frederic Bimbot. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1995)*, Detroit, MI, USA, May.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38.

Qing Dou and Kevin Knight. 2012. Large scale decipherment for out-of-domain machine translation. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Computational Language Learning (EMNLP-CoNLL 2012)*, pages 266–275, Jeju, Republic of Korea, July.

Qing Dou and Kevin Knight. 2013. Dependency-based decipherment for resource-limited machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1668–1676, Seattle, WA, USA, October.

Qing Dou, Ashish Vaswani, and Kevin Knight. 2015. Unifying bayesian inference and vector space models for improved decipherment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 836–845, Beijing, China, July.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 53–61, Sydney, Austrailia, July.

Sharon Goldwater and Thomas L. Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pages 744–751, Prague, Czech Republic, June.

Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 296–305, Prague, Czech Republic, June.

Yunsu Kim, Andreas Guta, Joern Wuebker, and Hermann Ney. 2016. A comparative study on vocabulary reduction for phrase table smoothing. In *Proceedings of the ACL 2016 1st Conference on Machine Translation (WMT 2016)*, pages 110–117, Berlin, Germany, August.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1995)*, Detroit, MI, USA, May.

Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. Unsupervised analysis for decipherment problems. In *Proceedings of the 2006 Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006)*, pages 499–506, Sydney, Austrailia, July.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 868–876, Prague, Czech Republic, June.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the 10th Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand, September.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 133–139, Philadelphia, PA, USA, July.

Sven Martin, Jörg Liermann, and Hermann Ney. 1998. Algorithms for bigram and trigram word clustering. *Speech Communication*, 24(1):19–37, April.

Malte Nuhn and Hermann Ney. 2014. EM decipherment for large vocabularies. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 759–764, Baltimore, MD, USA, June.

Malte Nuhn, Arne Mauser, and Hermann Ney. 2012. Deciphering foreign language by combining language models and context vectors. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 156–164, Jeju, Republic of Korea, July.

Malte Nuhn, Julian Schamper, and Hermann Ney. 2013. Beam search for solving substitution ciphers. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 1569–1576, Sofia, Bulgaria, August.

Malte Nuhn, Julian Schamper, and Hermann Ney. 2015. Unravela decipherment toolkit. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 549–553, Beijing, China, July.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March.

Sujith Ravi and Kevin Knight. 2011a. Bayesian inference for zodiac and other homophonic ciphers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 19–24, Portland, OR, USA, June.

Sujith Ravi and Kevin Knight. 2011b. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 12–21, Portland, OR, USA, June.

Sujith Ravi. 2013. Scalable decipherment for machine translation via hash sampling. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 362–371, Sofia, Bulgaria, August.

Christian Rishøj and Anders Søgaard. 2011. Factored translation with unsupervised word clusters. In *Proceedings of the 2011 EMNLP 6th Workshop on Statistical Machine Translation (WMT 2011)*, pages 447–451, Edinburgh, Scotland, July.

Julian Schamper. 2015. Unsupervised training with applications in natural language processing. Master's thesis, Computer Science Department, RWTH Aachen University, Aachen, Germany, September.

Joern Wuebker, Stephan Peitz, Felix Rietig, and Hermann Ney. 2013. Improving statistical machine translation with word class models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1377–1381, Seattle, USA, October.