Reducing Parameter Space for Word Alignment

Herve Dejean, Eric Gaussier, Cyril Goutte, and **Kenji Yamada**
Xerox Research Centre Europe
6, Chemin de Maupertuis, 38240 Meylan, France
{hdejean,gaussier,cgoutte,kyamada}@xrce.xerox.com

Abstract

This paper presents the experimental results of our attemps to reduce the size of the parameter space in word alignment algorithm. We use IBM Model 4 as a baseline. In order to reduce the parameter space, we pre-processed the training corpus using a word lemmatizer and a bilingual term extraction algorithm. Using these additional components, we obtained an improvement in the alignment error rate.

1 Introduction

We participated the workshop shared task for English-French and Romanian-English word alignment. We use IBM Model 4 as a baseline. The number of parameters in this model roughly scales as the product of the vocabulary sizes (ie number of types) in the source and target languages. In order to obtain better alignment performance, we wish to investigate techniques that may reduce the number of parameters, therefore increasing the data-to-parameter ratio. For that purpose, we preprocessed the training corpus using a word lemmatizer and a bilingual lexicon extraction algorithm. Section 2 briefly describes the base alignment algorithm, Section 3 describes our additional components, and Section 4 shows our experimental results, followed by Discussion and Conclusion in Section 5 and 6, respectively.

2 Word Alignment algorithm

We use IBM Model 4 (Brown et al., 1993) as a basis for our word alignment system. The model was implemented in a public software package GIZA++ (Och and Ney, 2000). We use default parameters provided with the package, namely, it was bootstrapped from Model 1 (five iterations), HMM model (five iterations) Model 3 (two iterations) and Model 4 (four iterations).

IBM Model 4 is a conditional generative model, which generates an English sentence (and a word alignment) given a foreign sentence (French or Romanian, in our experiments here). In the generative process, each English word e is duplicated ϕ times according to the probabilities given by the fertility table $n(\phi|e)$. Each duplicated English word e is then translated to a French (or Romanian) word f according to the probabilities given by the translation table $t(f|e)$. The position of f in the French sentence is then moved from the position of e in the English sentence by an offset δ. The probability of δ is given by the distortion table $d(\delta|A(e), B(f))$, which is conditioned on the word classes $A(e)$ and $B(f)$. In GIZA++, the word classes are automatically detected by a bilingual clustering algorithm.

The translation table $t(f|e)$ dominates the parameter space when the vocabulary size grows. In this paper, we focus on how to reduce the table size for $t(f|e)$. We apply two additional methods, lemmatization and bilingual lexicon extraction, described below. We expect two advantages by reducing the model parameter space. One is to reduce the memory usage, which allows us to use more training data. Another is to improve the data-to-parameter ratio, and therefore the accuracy of the alignment.

3 Reducing the Parameter Space

To reduce the model parameter space, we apply the following two methods. One is a rule-based word lemmatizer and another is a statistical lexical extraction algorithm.

3.1 Word Lemmatizer

We use a word lemmatizer program (XRCE, 2003) which converts words in variant forms into the root forms. We preprocess the training and the test corpora with the lemmatizer. Figure 1 and 2 show examples of how the lemmatizer works.

it would have **been** easy to say that these **sanctions** have to be **followed** rather than **making** them voluntary .
it would have **be** easy to say that these **sanction** have to be **follow** rather than **make** them voluntary .
il **aurait été** facile de dire que il **faut** appliquer **ces sanctions** à le lieu de **les** rendre **facultatives** .
il **avoir être** facile de dire que il **falloir** appliquer **ce sanction** à le lieu de **le** rendre **facultatif** .

Figure 1: Lemmatizer Example 1

this **is being done** to ensure that **our children** will receive a pension under the cpp .
this **be be do** to ensure that **we child** will receive a pension under the cpp .
cela **permettra** à **nos enfants** de pouvoir bénéficier de le régime de **pensions** de le canada .
cela **permettre** à **notre enfant** de pouvoir bénéficier de le régime de **pension** de le canada .

Figure 2: Lemmatizer Example 2

Applying the lemmatizer reduces the parameter space for the alignment algorithm by reducing the vocabulary size. Nouns (and adjectives for French) with different gender and number forms are grouped into the same word. Verbs with different tenses (present, past, etc.) and aspects (-ing, -ed, etc.) are mapped to the same root word. In particular, French verbs have many different conjugations: Some verb variants appear only once or twice in a corpus, and the statistics for those rare words are unreliable. Thus, we expect to improve the model accuracy by treating those variants as the same word.

On the other hand, there is a danger that lemmatization may lose useful information provided by the inflected form of a word. In particular, special words such as *do* and *be* may have different usage patterns for each variant (e.g., *done* vs. *doing*). In that case, lemmatization may actually hurt the performance.

3.2 Bilingual Lexical Extraction

Another additional component we use is a bilingual lexicon extraction algorithm. We run the algorithm over the same training data, and obtain a list of word translation pairs. The extracted word-pair list is used as an additional training data for GIZA++. This will give some bias for the alignment model parameters. This does not actually reduce the parameter space, but if the bias is taken to the extreme (e.g. some of the model parameters are fixed to zero), it will reduce the parameter space in effect.

For the bilingual lexicon extraction, we use a word alignment model different from IBM models. The purpose of using a different model is to extract 1-to-1 word translation pairs more reliably. The model (described below) assumes that a translation sentence pair is preprocessed, so that the pair is a sequence of content words. To select content words, we apply a part-of-speech tagger to remove non content words (such as determiners and prepositions). As the model focuses on the alignment of content words, we expect better performance than IBM models for extracting content word translation pairs.

We give here a brief description of the bilingual lexicon extraction method we use. This method takes as input a parallel corpus, and produces a probabilistic bilingual lexicon. Our approach relies on the word-to-word translation lexicon obtained from parallel corpora following the method described in (Hull, 1999), which is based on the word-to-word alignment presented in (Himstra, 1996).

We first represent co-occurrences between words across translations by a matrix, the rows of which represent the source language words, the columns the target language words, and the elements of the matrix the *expected alignment frequencies (EAFs)* for the words appearing in the corresponding row and column. Empty words are added in both languages in order to deal with words with no equivalent in the other language.

The estimation of the expected alignment frequency is based on the Iterative Proportional Fitting Procedure (IPFP) presented in (Bishop et al., 1975). This iterative procedure updates the current estimate $n_{ij}^{(k)}$ of the EAF of source word i with target word j, using the following two-stage equations:

$$n_{ij}^{(k,1)} = \sum_{s,(i,j)\in s} n_{ij}^{(k-1,2)} \times \frac{s_i}{n_{i}^{(k-1,2)}} \quad (1)$$

$$n_{ij}^{(k,2)} = \sum_{s,(i,j)\in s} n_{ij}^{(k,1)} \times \frac{s_j}{n_{.j}^{(k,1)}} \quad (2)$$

where $n_{i.}$ and $n_{.j}$ are the current estimates of the row and column marginals, s is a pair of aligned sentences containing words i and j, and s_i and s_j are the observed frequencies of words i and j in s. The initial estimates $n_{ij}^{(0,2)}$ are the observed frequencies of co-occurrences, obtained by considering each pair of aligned sentences and by incrementing the alignment frequencies accordingly. The sequence of updates will eventually converge and the EAFs are then normalized (by dividing each element n_{ij} by the row marginal $n_{i.}$), so as to yield probabilistic translation lexicons, in which each source word is associated with a target word through a score.

Using the bilingual lexicon thus obtained, we use a

	corpus size(E)	lem	vocab $\|E\|$	vocab $\|F\|$	Mem	Trial AER	Test AER	Test AERn
nolem-ef-1.2m	20M		57.7K	79.6K	993M	0.076	0.079	
nolem-ef-565k	10M		43.2K	59.7K	624M	0.090	0.085	0.213
nolem-ef-280k	5M		33.9K	46.8K	453M	0.081	0.089	0.221
nolem-ef-56k	1M		18.6K	25.3K	160M	0.141	0.107	
delem-ef-2-280k	5M	2	32.5K	43.3K	449M	0.076	0.093	
delem-ef-3-280k	5M	3	32.0K	42.1K	447M	0.087	0.092	
delem-ef-5-280k	5M	5	31.6K	41.4K	446M	0.097	0.092	
delem-ef-100-280k	5M	100	31.5K	41.2K	380M	0.087	0.088	
delem-ef-1000-280k	5M	1000	31.5K	41.2K	367M	0.077	0.088	
delem-ef-1000-56k	1M	1000	16.9K	22.2K	148M	0.130	0.103	
base-ef-1.2m	20M		44.5K	47.7K	571M			
base-ef-565k	10M		32.5K	33.9K	389M	0.102	0.162	0.290
base-ef-280k	5M		25.2K	26.1K	287M	0.123	0.167	
base-ef-56k	1M		13.9K	14.4K	112M	0.137	0.178	

Table 1: English-French shared task

	P_{th} (dup)	Trial AER	Test AER	Test AERn
nolem-er-56k		0.283	0.289	0.369
base-er-all		0.323	0.310	0.385
trilex-er-all-3	0.3	0.318	0.314	
trilex-er-all-2	0.2	0.313	0.313	
trilex-er-all-1	0.1	0.296	0.310	
trilex-er-all-05	0.05	0.282	0.310	
trilex-er-all-02	0.02	0.297	0.308	
trilex-er-all-01	0.01	0.286	0.307	0.382
trilex-er-all-01-2	0.01 (2)	0.281	0.302	
trilex-er-all-01-5	0.01 (5)	0.282	0.298	0.374
trilex-er-all-01-10	0.01 (10)	0.283	0.295	
trilex-er-all-01-20	0.01 (20)	0.296	0.297	
trilex-er-all-01-50	0.01 (50)	0.293	0.300	

Table 2: Romanian-English shared task

simple heuristic, based on the best match criterion described in (Gaussier et al., 2000) to align lexical words within sentences. We then count how many times two given words are aligned in such a way, and normalize the counts so as to get our final probabilistic translation lexicon.

4 Experiments

4.1 English-French shared task

In the English-French shared task, we experimented the effect of the word lemmatizer. Table 1 shows the results.[1]

[1]In the table, AER stands for Average Error Rate without null-aligned words, and AERn was calculated with null-aligned words. See the workshop shared-task guideline for the definition of AER. Mem is the memory requirement for running GIZA++.

Due to our resource constraints, we used only a portion of the corpus provided by the shared task organizer. Most of our English-French experiments were carried out with the half (10 million) or the quarter (5 million) of the training corpus. We ran three different systems (nolem, base, and delem) with some different parameters. The system nolem is a plain GIZA++ program. We only lowercased the training and the test corpus for nolem. In base and delem, the corpus were preprocessed by the lemmatizer. In base system, the lemmatizer was applied blindly, while in delem, only rare words were applied with lemmatization.

As seen in Table 1, applying the lemmatizer blindly (base) hurt the performance. We hypothesized that the lemmatizer hurts more, when the corpus size is bigger. In fact, the Trial AER was better in base-ef-56k than nolem-ef-56k. Then, we tested the performance when

we lemmatized only rare words. We used word frequency threshold to decide whether to lemmatize or not. For example, delem-ef-2-280k lemmatized a word if it appeared less than twice in the training corpus. In general, the selective lemmatization (delem-ef-*-280k) works better than complete lemmatization (base-ef-280k). In some thresholds (delem-ef-{100,1000}-280k), the Test AER was slightly better than no lemmatization (nolem-ef-280k). However, from this experiment, it is not clear where we should set the threshold. We are now investigating this issue.

4.2 Romanian-English shared task

In the Romanian-English shared task, we experimented how the bilingual lexicon extraction method affects the performance. Table 2 shows the results.

We have three systems nolem, base, and trilex for this task. The first two systems are the same as the English-French shared task, except we use a lemmatizer only for English.[2] The system trilex uses additional bilingual lexicon for training GIZA++. The lexicon was extracted by the algorithm described in 3.2. We tried different thresholds P_{th} to decide which extracted lexicons are used. It is an estimated word translation probability given by the extraction algorithm. We also tested the effect of duplicating the additional lexicon by 2, 5, 10, or 20 times, to further bias the model parameters.

As our extraction method currently assumes word lemmatization, we only compare trilex results with base systems. As seen in the Table 2, it performed better when the extracted lexicons were added to the training data (e.g., base-er-all vs. trilex-er-all-01). The lexicon duplication worked best when the duplication was only twice, i.e. duplicating additional lexicon too much hurt the performance. For the threshold P_{th}, it worked better when it was set lower (i.e., adding more words). Due to the time constraints, we didn't test further lower thresholds.

5 Discussion

As we expected, the lemmatizer reduced the memory requirement, and improved the word alignment accuracy when it was applied only for infrequent words. The behavior of using different threshold to decide whether to lemmatize or not is unclear, so we are now investigating this issue.

Adding extracted bilingual lexicons to the training data also showed some improvement in the alignment accuracy. Due to our experimental setup, we were unable carry this experiment with selective lemmatization. We are going to try such experiment pretty soon.

[2]We do not have a Romanian lemmatizer, but we used a part-of-speech tagger by Dan Tufis for Romanian to extract bilingual lexicon.

6 Conclusion

We presented our experimental results of the workshop shared task, by using IBM model 4 as a baseline, and by using a word lemmatizer and a bilingual lexicon extraction algorithm as additional components. They showed some improvement over the baseline, and suggests the need of careful parameter settings.

Acknowledgment

We are grateful to Dan Tufis for a Romanian corpus preprocessed with his Romanian part-of-speech tagger. This research was supported by the European Commission under the TransType2 project no. IST-2001-32091.

References

Brown P, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2).

Bishop S. Fiendbeg and P. Holland. 1975. *Discrete Multivariate Analysis*, MIT-Press.

Gaussier E, D. Hull, and S. Ait-Mokhtar. 2000. Term Alignment in Use: Machine-Aided Human Translation. In J. Veronis, Ed. *Parallel Text Processing – Alignment and Use of Translation Corpora*. Kluwer Academic Publishers.

Hiemstra D. 1996. *Using Statistical Methods to Create a Bilingual Dictionary*. Master Thesis. Universiteit Twente.

Hull D. 1999. Automating the construction of bilingual terminology lexicons. *Terminology*, 4(2).

Och F. J. and H. Ney. 2000. Improved Statistical Alignment Models. *ACL-00*.

Xerox Research Centre Europe. 2003. Finite-State Linguistic Components. CA Linguistic Technology: Demos. `http://www.xrce.xerox.com/competencies/content-analysis/toolhome.en.html`.