Gap Between Theory and Practice: Noise Sensitive Word Alignment in Machine Translation

Tsuyoshi Okita[1] TOKITA@COMPUTING.DCU.IE
Yvette Graham[2] YGRAHAM@COMPUTING.DCU.IE
Andy Way[1] AWAY@COMPUTING.DCU.IE
Dublin City University, {CNGL[1],NCLT[2]} / School of Computing

Editor: Editor's name

Abstract

Word alignment is to estimate a lexical translation probability $p(e|f)$, or to estimate the correspondence $g(e, f)$ where a function g outputs either 0 or 1, between a source word f and a target word e for given bilingual sentences. In practice, this formulation does not consider the existence of 'noise' (or outlier) which may cause problems depending on the corpus. N-to-m mapping objects, such as paraphrases, non-literal translations, and multi-word expressions, may appear as both noise and also as valid training data. From this perspective, this paper tries to answer the following two questions: 1) how to detect stable patterns where noise seems legitimate, and 2) how to reduce such noise, where applicable, by supplying extra information as prior knowledge to a word aligner.
Keywords: Probability density estimation problem, Noise.

1. Introduction

N-to-m mapping objects, such as paraphrases, non-literal translations, and multi-word expressions, appear as both noise and as valid training data for word alignment (Brown et al., 1993; Och and Ney, 2003; Taskar et al., 2005) in Machine Translation. It is often the case that noisy data has a negative effect, as even small numbers of outliers may severely decrease the overall performance and the removal makes the performance better after detecting them, by novelty detection or outlier detection. In contrast to this, in our case, removal of a small amount of such noisy data improves performance while too much removal deteriorates performance; this phenomenon is nonlinear and noise is not Gaussian type. This is caused by the definition of word alignment (Definition 4 in Appendix), as it seeks the probability density function of a word e given a word f (or a probability of 1-to-n mapping objects); this definition omits how to handle difficult cases in real life data such as n-to-m mapping objects such as in Figure 1, and rely on phrase extraction heuristics (Och and Ney, 2003) to recover considerable numbers of n-to-m mapping objects. A typical difficulty of this problem is that in word alignment we often cannot use the immediate evaluation measure due to the unavailability of a hand annotated corpus with alignment links. Furthermore, even in the case when we can use such corpus, the better quality of word alignment which is measured by AER (Alignment Error Rate) (Och and Ney, 2003) is empirically shown by various researchers that it does not often result in the better translation quality (Fraser and Marcu, 2007).

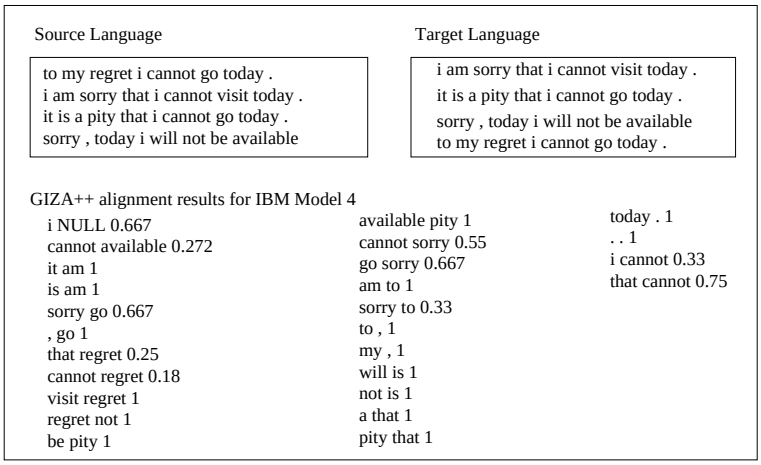

Figure 1: An example alignment of paraphrases in a monolingual case: a training corpus consists of four sentence pairs. Results show that only the matching between the colon is correct

Note that PB-SMT does not require $P(e|f)$ but $P(\bar{e}|\bar{f})$ where $\bar{e}$ denotes an English phrase and e denotes an English word. Although phrase alignment of Marcu and Wong (2002) aims at this $P(\bar{e}|\bar{f})$, this approach is often infeasible due to its computational cost. A word alignment approach considerably reduces this cost by neglecting correlation between neighboring variables, which creates the problem of n-to-m mapping objects by this compromise.

We use the following notation: e denotes an English word, f denotes a Foreign word, $\breve{e}$ denotes an English sentence, $\bar{e}$ denotes an English phrase, $|\breve{e}_i|$ denotes a sentence length of $\breve{e}_i$, and $\breve{e}_{\bar{i}}$ denotes a reference translation of correspondent Foreign sentence $\breve{f}_i$.

2. Our Methods

Given that a parallel corpus contains unlabeled n-to-m mapping objects (the usual case in Machine Translation), from the above observation about n-to-m mapping objects, our first interest is to detect such objects. What we would like to solve is the following problem:

Definition 1 (n-to-m mapping objects detection) *Let $S = \{(\breve{e}_1, \breve{f}_1), \ldots, (\breve{e}_n, \breve{f}_n)\}$ be a parallel corpus. For a given parallel corpus, the n-to-m mapping objects detection task is to detect sentence pairs $(\breve{e}_i, \breve{f}_i)$ which include n-to-m mapping objects, such as paraphrases, non-literal translations, and multi-word expressions.*

There are two ways to solve this: extrinsically and intrinsically. The extrinsic method does not detect sentences that include n-to-m mapping objects in a straight forward way, but instead detects them via indirect measures that identify sentences that contain n-to-m mapping objects. However, in this case, due to the nature of indirectness, we need to enlarge our objectives: we enlarge n-to-m mapping objects to include noise (or unfavorable elements). Hence, we solve the following noise detection task:

Definition 2 (Noise detection) *Let $S = \{(\breve{e}_1, \breve{f}_1), \ldots, (\breve{e}_n, \breve{f}_n)\}$ be a parallel corpus consisting of a training and development corpus. For a given parallel corpus, a noise detection task is to detect sentence pairs $(\breve{e}_i, \breve{f}_i)$ that contain noise (or unfavorable elements). Note that unfavorable elements include n-to-m mapping objects.*

The intrinsic method identifies n-to-m mapping objects themselves. However, n-to-m mapping objects include at least three cases such as paraphrases, non-literal translations, and multi-word expressions in this case, and how to tackle these three cases is not straight forward and requires further investigation. In here, we limit it to the Multi-Word Expression (MWE) detection task which is defined as follows:

Definition 3 (MWE detection) *Let $S = \{(\breve{e}_1, \breve{f}_1), \ldots, (\breve{e}_n, \breve{f}_n)\}$ be a parallel corpus consisting of a training and development corpus. For a given parallel corpus, a MWE detection task detects MWEs in each sentence $(\breve{e}_i, \breve{f}_i)$.*

On the one hand, once we have detected sentences that contain noise, we need a method of dealing with such sentences for word alignment. On the other hand, when we detect MWEs in the training corpus, we need to deal with such alignment links. In sum, we consider the following two methods:

1. Detect sentences including n-to-m mapping objects, remove such sentences and run a word aligner with the reduced training set.

2. Run a MWE extractor, incorporate the detected MWEs into a word aligner as a prior knowledge, and run the word aligner.

3. Details of Our Methods

3.1. Method 1: Reduced Training Set After Noise Detection

Noise Detection We mention in Definition 2 how to detect sentence pairs which include noise. One problem is that we have no appropriate direct measure to assess the quality of word alignment due to the unavailability of hand annotated corpus with alignment links: AER (Alignment Error Rate) (Och and Ney, 2003) can be applied only if a hand annotated corpus exists and if in-domain test data is available. Although an indirect measure such as perplexity gives an indication for assessing the progress of EM training (Dempster et al., 1977; McLachlan and Krishnan, 1997), this does not give appropriate indication of the success of word alignment since perplexity is a simple transformation of the cross entropy from information theory, that does not use any information about n-to-m mapping objects. In the end-to-end setting of Machine Translation, we can use the distance measure which is defined a priori, an evaluation measure, such as BLEU_n for example.

In sum, the approach taken in Okita (2009) is as follows. Let $S = \{(\breve{e}_1, \breve{f}_1), \ldots, (\breve{e}_n, \breve{f}_n)\}$ be a training corpus and let $M : \breve{f} \rightarrow \breve{e}$ be our MT system trained on this training corpus S. If the distance between a reference translation $\breve{e}_{\bar{i}}$ and $M(\breve{f}_i)$ is big for relatively small data sets, this may indicate that the sentence $\breve{f}_i$ is relatively difficult to translate; this may be due to a training sentence $\breve{f}_i$ too complex for the model complexity of MT system M. Further details can be found in Okita (2009).

Word Alignment Based on Reduced Sentences By removing the detected noisy sentences we reduce the training corpus and rerun the word aligner[1].

3.2. Method 2: Supply Prior Knowledge After MWE Detection

MWE Detection One way to extract MWEs in unidirectional way is based on Kupiec. (1993) where linguistic knowledge about typical POS patterns are available, e.g. for Noun phrases in French: N N, N prep N, and N Adj. However, crucial difference is that after we extracted MWEs in bidirectional way, our method takes intersection of them.

Incorporation of Knowledge About MWEs into Word Alignment The EM algorithm-based word aligner uses maximum likelihood in its M-step. Our method replaces this maximum likelihood estimate with the MAP (Maximum A Posteriori) estimate, which is a basic Bayesian machine learning method. Let t be a lexical translation probability $t(e|f)$; note that often t is omitted in word alignment literature but for our purposes this needs to be explicit.

$$\begin{aligned}
\mathbf{E^{EXH}}: &\quad q(z;x) = p(z|x;\theta) \\
\mathbf{M^{MLE}}: &\quad t' = \arg\max_t Q(t, t^{old}) = \arg\max_t \sum_{x,z} q(z|x) \log p(x,z;t) \\
\mathbf{M^{MAP}}: &\quad t' = \arg\max_t Q(t, t^{old}) + \log p(t) = \arg\max_t \sum_{x,z} q(z|x) \log p(x,z;t) + \log p(t).
\end{aligned}$$

Then, the prior $\log p(t)$, a probability used to reflect the degree of prior belief about the occurrences of the events, can embed prior knowledge about MWEs.

Table 1 shows two example phrase pairs for French to English *c'est la vie* and *that is life*, and *la vie en rose* and *rosy life* with the initial value for the EM algorithm, the prior value and the final lexical translation probability for GIZA++ IBM Model 4 and that of our modified GIZA++. GIZA++ achieves the correct result when anchor words 'life' and '*vie*' are used to assign a value to the prior in our model.

A prior for IBM Model 1 considers all possible alignments exhaustively in E-Step as in the definition of EM algorithm (while IBM Model 3 and 4 only sample a neighborhood alignments around the best alignment). Let us give information about alignment link between e and f by $T = \{(sentID, t_i, t_j, pos_i, pos_j), \ldots, \}$ into prior. The prior $p(t) = p(t; e, f, T)$ for given word e and f in a sentence is defined simply 1 if they have alignment link, 0 if they are not connected, and uniform if their link is not known:

$$p(t) = p(t; e_i, f_i, T) \quad = \quad \begin{cases} 1 & (e_i = t_i, f_j = t_j) \\ 0 & (e_i = t_i, f_j \neq t_j) \\ 0 & (e_i \neq t_i, f_j = t_j) \\ \text{uniform} & (e_i \neq t_i, f_j \neq t_j) \end{cases}$$

Then we embed this prior in the M-step of EM algorithm where we replaced its likelihood estimate with MAP estimate (Okita et al., 2010). Although this is for the case of IBM

1. It is to be noted that we set aside the problem of whether this approach actually improves on the test set accuracy.

pair	GIZA++(no prior)			Ours(with prior)		
EN-FR	fin	ini	prior	fin	ini	prior
is *NULL*	1	.25	0	0	.25	.25
rosy *en*	1	.5	0	0	.5	.2
that .	1	.25	0	0	.25	.25
life *la*	1	.25	0	0	.25	0
. *c'*	1	.25	0	0	.25	.25
that *c'*	0	.25	0	1	.25	.25
is *est*	0	.25	0	1	.25	.25
life *vie*	0	.5	0	1	.5	1
rosy *rose*	0	.25	0	1	.25	.2

Table 1: The benefit of prior knowledge about anchor words is illustrated by toy data. Given two sentence pairs {(that is life . , *c' est la vie* .), (rosy life, *la vie en rose*)} and anchor words $\{(1, \text{life}, vie, 3, 4), (2, \text{life}, vie, 2, 2)\}$, we compare the results of GIZA++ with IBM Model 4 and that of our modified GIZA++. The columns, labeled with *ini*, *fin* and *prior*, show respectively the final lexical probability $t(f|e)$, the initial value for the EM algorithm, and the prior value explained in this paper. Notice that all the links are incorrect in GIZA++ while all the links are correct in ours.

Model 1, IBM Models 3 and 4 are essentially the same except that they are not proper. Due to the space problems in here, further details can be found in Okita et al. (2010).[2]

4. Experimental Results

Our baseline is a standard log-linear PB-SMT system based on Moses. The GIZA++ implementation (Och and Ney, 2003) of IBM Model 4 is used for word alignment. For phrase extraction the grow-diag-final heuristics described in Och and Ney (2003) is used to derive the refined alignment. We then perform MERT process which optimizes the BLEU metric, while a 5-gram language model is derived with Kneser-Ney smoothing trained with SRILM on the English side of the training data. We use Moses for decoding. Our implementation for Method 2 is based on GIZA++.

We use NTCIR-8 patent corpus for EN–JP (Fujii et al., 2010) and Europarl corpus for EN–FR (Koehn, 2005). We randomly select 50k and 200k sentence pairs as training corpus. For EN–JP patent corpus, we use 1k sentence development set and NTCIR-8 test set. For EN–FR Europarl corpus, we use dev2006 and test2006. The results of Method 1 and 2 are shown in Table 2. The best improvement by Method 1 was 0.82 BLEU points absolute for 200k EN–JP, while that by Method 2 was 0.97 BLEU points absolute for 50k EN–JP. However, for 50k JP–EN, Method 1 improves only 0.10 BLEU point and Method 2 did not

2. Note that in practice IBM Model 4 is required due to its quality. However, the lower-order IBM Models are required for better initialization parameters: in order to obtain the result of IBM Model 4, we perform 5 iterations of Model 1, HMM, Models 3 and 4, iteratively.

improve. These differences are probably due to the amount of noise in parallel corpus: we experienced in IWSLT 09 that some corpus has less noise (or a lot of redundant sentence pairs) from the beginning which does not require the 'noise' treatment (Ma et al., 2009).

size	lang	system	BLEU	size	lang	system	BLEU
50k	EN–JP	baseline	16.33	50k	JP–EN	baseline	22.11
50k	EN–JP	Method 1	16.99	50k	JP–EN	Method 1	22.22
50k	EN–JP	Method 2	17.30	50k	JP–EN	Method 2	22.11
200k	EN–JP	baseline	23.42	200k	JP–EN	baseline	21.68
200k	EN–JP	Method 1	24.24	200k	JP–EN	Method 1	22.93
200k	EN–JP	Method 2	24.22	200k	JP–EN	Method 2	22.45
50k	FR–EN	baseline	17.68	50k	EN–FR	baseline	17.80
50k	FR–EN	Method 1	17.90	50k	EN–FR	Method 1	18.30
50k	FR–EN	Method 2	17.81	50k	EN–FR	Method 2	18.02
200k	FR–EN	baseline	18.40	200k	EN–FR	baseline	18.20
200k	FR–EN	Method 1	18.85	200k	EN–FR	Method 1	18.62
200k	FR–EN	Method 2	18.99	200k	EN–FR	Method 2	18.60

Table 2: Results for Method 1 and 2.

5. Conclusion and Further Works

Word alignment is to estimate a lexical translation probability $p(e|f)$ between a source word f and a target word e for given bilingual sentences. This paper presented a robust method for word alignment under the existence of n-to-m mapping objects. Since n-to-m mapping objects pose two different challenges for word alignment, noise (or outlier), as well as valid training data, this situation is not just an application of outlier detection in pattern analysis. Method 1 detects sentences which include n-to-m mapping objects and reduces these sentences, while Method 2 detects MWEs and incorporates detected alignment link information into word alignment. The best improvement by Method 1 was 0.82 BLEU points absolute for 200k EN–JP, while that by Method 2 was 0.97 BLEU points absolute for 50k EN–JP.

There are several further works. Firstly, we consider MWEs in Method 2 but we did not consider paraphrases. This extension requires different extraction method but once we obtain the paraphrases we can reuse the mechanism shown in here. Secondly, if a corpus has already a lot of redundant sentences such as in IWSLT 09 ZH–EN corpus, Method 1 might not work since we notice that the duplication has the same effect as Method 1.[3] It would be convenient if we can measure the applicability before we apply Method 1. Thirdly, although there are several discriminative approaches to word alignment exist (Taskar et al., 2005), they require small hand annotated corpus with alignment links. The mechanism of Yu and Joachims (2009) may allow us to implement word alignment in a discriminative way

3. Empirically, we checked in several cases that instead of removing detected sentences as in Method 1, we duplicated 5 times of detected sentences. Unless GIZA++ crashes, the performance of duplication method seemed to be comparable.

without the need of such hand annotated corpus. However, this way of implementing word alignment has difficulty in incorporating the mechanism of IBM Model 3 and 4.

Acknowledgments

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (`http://www.cngl.ie`) at Dublin City University. We would also like to thank the Irish Centre for High-End Computing.

References

P. F. Brown, V.J.D. Pietra, S.A.D. Pietra, and R.L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics, Vol.19, Issue 2*, pages 263–311, 1993.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, pages 1–38, 1977.

A. Fraser and D. Marcu. Measuring word alignment quality for statistical machine translation. *Computational Linguistics, Squibs and Discussion*, 33(3):293–303, 2007.

A. Fujii, M. Utiyama, M. Yamamoto, T. Utsuro, T. Ehara, H. Echizen-ya, and S. Shimohata. Overview of the patent translation task at the NTCIR-8 workshop. *In Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access*, pages 293–302, 2010.

P. Koehn. Europarl: A parallel corpus for statistical machine translation. *In Proceedings of the Machine Translation Summit*, pages 79–86, 2005.

J. Kupiec. An algorithm for finding Noun phrase correspondences in bilingual corpora. *In Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL 1993)*, pages 17–22, 1993.

Y. Ma, T. Okita, O. Cetinoglu, J. Du, and A. Way. Low-resource Machine Translation using MaTrEx: the DCU Machine Translation system for IWSLT 2009. *In Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2009)*, pages 29–36, 2009.

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

G.J. McLachlan and T. Krishnan. The EM algorithm and extensions. *Wiley Series in probability and statistics*, 1997.

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

T. Okita. Data cleaning for word alignment. *In Proceedings of Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009) Student Research Workshop*, pages 72–80, 2009.

T. Okita, A. Maldonado Guerra, Y. Graham, and A. Way. Multi-Word Expression-sensitive word alignment. *In Proceedings of the Fourth International Workshop On Cross Lingual Information Access (CLIA2010, collocated with COLING2010)*, pages 1–8, 2010.

B. Taskar, S. Lacoste-Julien, and D. Klein. A discriminative matching approach to word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, pages 73–80, Vancouver, British Columbia, Canada, October 2005.

C.N. Yu and T. Joachims. Learning structural SVMs with latent variables. *In Proceedings of the International Conference on Machine Learning (ICML 2009)*, pages 1169–1176, 2009.

Appendix A. Word Alignment Task, Phrase Extraction Heuristics and Noisy Channel Model

Definition 4 (Word Alignment Task) *We are given a pair of sentence aligned bilingual texts* $S = \{(\breve{f}_1, \breve{e}_1), \ldots, (\breve{f}_n, \breve{e}_n)\}$*, where* $\breve{f}_i = (\bar{f}_{i,1}, \ldots, \bar{f}_{i,|f_i|})$ *and* $\breve{e}_i = (\bar{e}_{i,1}, \ldots, \bar{e}_{i,|e_i|})$*. The task of word alignment is to find a lexical translation probability* $p_{f_i} : e_i \rightarrow p_{f_j}(e_i)$ *such that* $\Sigma p_{f_j}(e_i) = 1$ *and* $\forall e_i : 0 \leq p_{f_j}(e_i) \leq 1$ *(It is noted that some models such as IBM Model 3 and 4 have deficiency problems). Note that IBM Models introduce an alignment / distortion function as latent variable to solve this problem as a missing value problem.*

Definition 5 (Phrase Extraction) *The phrase extraction algorithm extracts all consistent phrase pairs from a word aligned sentence pair (Och and Ney, 2003).*

Definition 6 (Bayesian Noisy Channel Model) *We assume that sentence pairs* $(\breve{e}, \breve{f})$ *are drawn i.i.d. (independent and identically distributed) according to the fixed (but unknown) underlying distributions* $p(\bar{f}|\bar{e})p(e)$*. Then, for a given test sentence* $\breve{f}$*, our task is to obtain a sentence* $\underline{\breve{e}}$ *which maximizes the following problem:*

$$\begin{cases} \underline{\breve{e}} = & \arg\max_{e \in E'} p(\bar{f}|\bar{e})p(e) \qquad \textit{(decoding task)} \\ & \textit{such that} \begin{cases} |\hat{p}(\bar{f}|\bar{e}) - p(\bar{f}|\bar{e})| \leq \delta_1 & \textit{(phrase alignment task)} \\ |\hat{p}(e) - p(e)| \leq \delta_2 & \textit{(language modeling task)} \end{cases} \end{cases}$$

where $p(\bar{f}|\bar{e})$ *denotes the target probability of phrase alignment task,* $p(e)$ *denotes the target probability of language modeling task (up to Markov order n; typical n is around 5),* $\hat{p}(\bar{f}|\bar{e})$ *denotes the true probability of phrase table,* $\hat{p}(e)$ *denotes the true probability of language model, and* $||$ *denotes some distance measure between two probability densities.*