WORD ALIGNMENT BASED ON MULTI-GRAIN MODEL

Yanqing He, Yu Zhou and Chengqing Zong

NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190

ABSTRACT

Word alignment plays a critical role in statistical machine translation (SMT) and cross-language information retrieval. Until now, most existing methods get the word alignment within the whole range of the sentence length. The alignment quality is unsatisfactory. In this paper, we propose a novel approach to word alignment based on multi-grain model (WAMG). We split a parallel sentence pair into blocks in different grain and get the word alignments within each corresponding block. Our approach is able to restrict the search space of word alignment in the relatively accurate local range and reduce the mapping error. The experiments have shown that our approach outperforms the traditional word alignment algorithm relatively by about 12% in AER and improves the performance of Chinese-to-English translation system relatively by about 2.8% in BLEU.

Index Terms — Statistical machine translation, word alignment, multi-grain, sub-sentence, block

1. INTRODUCTION

Word alignment, which maps source sentence words to target sentence words, is a vital component of statistical machine translation (SMT) and cross-language information retrieval. The quality of word alignment greatly contributes to the performance of a SMT system.

Many approaches for word aligning between parallel texts have been proposed. Some statistical models treat the word alignment as a hidden process, and employ probabilistic models to obtain the word alignment [1][2][3]. Reference [4] introduces a log-linear combination of IBM models and HMM model. Some heuristic models obtain word alignment by using various similarity functions between bilingual word pairs [5][6][7]. Some supervised approaches treat word alignment as a statistical classification problem, which make use of the conditional random fields (CRF) model [8], maximum entropy model [9][10], large margin model [11], or neural networks model [12] and so on. However, all the methods above obtain the word alignment within the whole range of the sentence length, which entails poor alignment quality, especially when the sentence is long.

Figure 1 gives an example of word alignment. (a) is a result of word alignment using traditional approach GIZA++, which is obtained in the whole range of a parallel Chinese-English sentence pair. For the Chinese sentence $c_1^5 = c_1 c_2 c_3 c_4 c_5$ and the English sentence $e_1^6 = e_1 e_2 e_3 e_4 e_5 e_6$, there are 7 links. Each link stands for the corresponding relation between a Chinese word and an English word. A solid line denotes a correct link and a dotted line denotes a wrong one. In Figure 1(a), the Chinese word c_4 and English word e_2 are incorrectly aligned. If we segment the sentence pair into two blocks $\{(c_1 c_2 c_3) \Leftrightarrow (e_1 e_2 e_3)\}$ and $\{(c_4 c_5) <=> (e_4 e_5 e_6)\}$, shown as (b) in Figure 1 (separated with bold line), the word alignment could be found within each block. The word c_4 has no possibility to be aligned to e_2. Here a contiguous source string is aligned to a contiguous target string to generate a block.

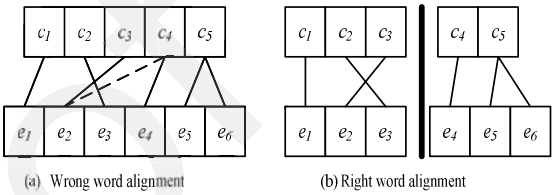

Figure 1: An example of word alignment.

Inspired by the idea to split a parallel sentence pair into blocks, we propose a new approach of word alignment based on multi-grain model (WAMG). The grain means the size of the block. WAMG includes three steps: a) Under different levels of grain, segmenting the input sentence pair into sub-sentences, aligning and merging these sub-sentence pairs into blocks. The input sentence pairs of current grain are the blocks of last grain; b) Under each grain obtaining word alignment within each block and combining them into the word alignment of the whole sentence; c) Linearly weighting these word alignments under different levels of grain to get the final word alignment of the whole sentence.

Remainder of this paper is organized as follows: Section 2 gives the definition of multi-grain model for word aligning. Section 3 describes the details of getting word alignment by WAMG. The experimental results and analysis are given in Section 4. Finally we will draw our conclusions and propose future work in Section 5.

2. WAMG: WORD ALIGNMENT BASED ON MULTI-GRAIN MODEL

Given a parallel sentence pair (c_1^J, e_1^I), the source language sentence is $c_1^J = c_1 .. c_j .. c_J$, and the target language sentence is $e_1^I = e_1 .. e_i .. e_I$. Equation $a_j = i$ means the source word c_j is aligned to the target word e_i. We define the alignment problem as finding the alignment of the words within the sentence pair $A = a_1^J = a_1 \ldots a_j \ldots a_J$ that maximizes the probability $\Pr(A \mid c_1^J, e_1^I)$ given c_1^J and e_1^I.

We introduce S (sub-sentence alignment) as a hidden process. Given some features to segment the original source sentence into

M sub-sentences (continuous string in the sentence) and the original target sentence into N sub-sentences, we denote by $c_1^J = sc_1^M$ and $e_1^I = se_1^N$ where sc_m is the m-th sub-sentence in c_1^J and se_n is the n-th sub-sentence in e_1^I. $s_m = n$ means the m-th source sub-sentence is aligned to the n-th target sub-sentence. So we get $S = s_1^M$.

It is easy to see that there are 6 types of sub-sentence alignment: 1-0, 0-1, 1-1, 1-m, n-1, and m-n. In the alignment types of 1-m, n-1 and m-n, m and n denote the sub-sentences' number that is more than one. These sub-sentences may be adjacent or not. We merge 1-m, n-1 and m-n into 1-1 alignment. The alignment types of 1-0 and 0-1 are abandoned for they are unable to provide any alignment information. Then all types of sub-sentence alignment are turned into blocks. $c_1^J \Rightarrow bc_1^R$ [1] and $e_1^I \Rightarrow be_1^R$ where bc_r and be_r respectively denote the Chinese side and the English side in the r-th block. So the original sentence pair is recombined into R blocks and $\{(bc_r) \Leftrightarrow (be_r)\}$ is the r-th block.

The word alignment of (c_1^J, e_1^I) is modeled as Equation (1). Here, A^r means the word alignment within the block $\{(bc_r) \Leftrightarrow (be_r)\}$. $p(S \mid c_1^J, e_1^I)$ models the process of finding the sub-sentence alignment in the whole sentence pair and $p(A^r \mid bc_r, be_r)$ denotes the process of finding the word alignment within the r-th block.

$$\begin{aligned} P(A \mid c_1^J, e_1^I) &= \sum_S p(S, A \mid c_1^J, e_1^I) \\ &= \sum_S p(S \mid c_1^J, e_1^I) p(A \mid c_1^J, e_1^I, S) \\ &= \sum_S p(S \mid c_1^J, e_1^I) p(A \mid sc_1^M, se_1^N, S) \\ &= \sum_S p(S \mid c_1^J, e_1^I) p(A \mid bc_1^R, be_1^R) \\ &= \sum_S p(S \mid c_1^J, e_1^I) \prod_{r=1}^{R} p(A^r \mid bc_r, be_r) \end{aligned} \quad (1)$$

Accordingly, we get the following decision rule:

$$\begin{aligned} \hat{A} &= \arg\max_A \{P(A \mid c_1^J, e_1^I)\} \\ &= \arg\max_A \{\sum_S p(S \mid c_1^J, e_1^I) \times p(A \mid sc_1^M, se_1^N, S)\} \\ &\approx \arg\max_S p(S \mid c_1^J, e_1^I) \times \prod_{r=1}^{R} \arg\max_{A^r} \{p(A^r \mid bc_r, be_r)\} \end{aligned} \quad (2)$$

Here we use the maximum probability of sub-sentence alignment to approximate the best sub-sentence alignment and in each block we only find the best word alignment.

In order to trade off the recall and precision, different types of features are introduced into our model and we can get different sub-sentence alignments with different features to segment sentence pair. Provided we use K levels of grains in all, then let S_k denote the sub-sentence alignment in segmentation of the k-th grain and A_k means the word alignment of (c_1^J, e_1^I) from S_k. $P_k(A_k \mid c_1^J, e_1^I)$ models the probability of word alignment of (c_1^J, e_1^I) in segmentation of the k-th grain. A linear model is used to incorporate all the word alignments in different grains to find our final word alignment of (c_1^J, e_1^I).

$$\hat{A} = \arg\max_A \{\sum_{k=1}^{K} \lambda_k P_k(A_k \mid c_1^J, e_1^I)\} \quad (3)$$

Here λ_k is the weight of $P_k(A_k \mid c_1^J, e_1^I)$.

3. OBTAINING WORD ALIGNMENT WITH WAMG

In order to explain how to acquire the word alignment of the whole sentence pair by our WAMG, we illustrate the process in Figure 2.

There are K levels of grains. In each grain, we choose some features to segment the source and target sentence into sub-sentences. By using the matching score algorithm of the current grain we match and combine these sub-sentence pairs to obtain the bilingual blocks. In order to deeply employ more segmenting features and acquire more information from local word alignment, we consider the blocks of last grain as our input sentence pairs of current grain. Then we train word alignment within each block. Here we can use any word alignment model mentioned in Section 1, such as statistical or heuristic or supervised models. In our experiments we choose GIZA++ to obtain the word alignment in each block. Then we get each word alignment of the whole sentence pair under each grain of segmentation. The word alignments under all levels of grains are called as grain-based word alignments. Finally a linear model is used to combine the grain-based word alignments to find the final word alignment of whole sentence pair.

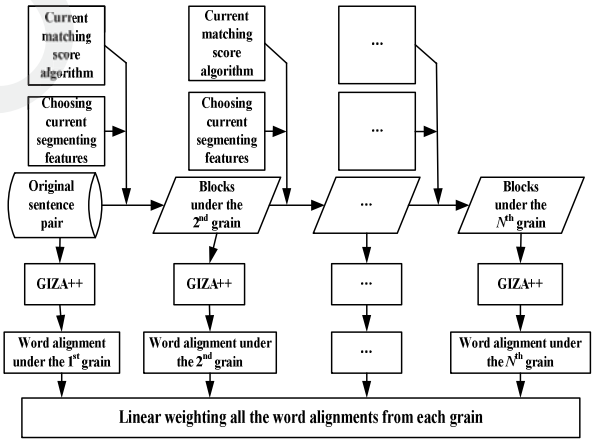

Figure 2: The training procedure of our WAMG.

In our experiments we employ 3 levels of grain. In each grain we use a step-extended-feature algorithm to match the source and target sub-sentence. In Section 3.2 we give the matching score algorithm. In each grain the matching score algorithms are the same except for the segmenting features which are used to segment sentences into sub-sentences and the scoring features which are used to compute the score between sub-sentence pairs.

3.1 The Segmenting Features and the Scoring Features of Each Grain

Under 1st grain we use the original sentence and the bilingual block is the original sentence pair. For the 2nd grain we choose the

[1] Here $\Rightarrow$ means the sentence on the left is composed of the sub-sentences in the blocks on the right without the consideration of their order and the abandonment of the sub-sentence aligned to NULL.

following six punctuations: 。 ！ ？ ， ： ； in Chinese sentence, and . ! ? , : ; in English sentence as our features to segment the input sentence pair (the blocks of 1st grain). For the 3rd grain, we use the key words as segmentation anchors. The key words include the conjunction (e.g. 'but', 'if', 'though', etc.) and some interrogative words (e.g. 'who', 'what', 'which', 'why', etc).

In each grain, we use the following three features to score the bilingual sub-sentences:

(1) IBM dictionary: The IBM dictionary is obtained from the GIZA++ dictionary of the last grain. In each direction of the GIZA++ dictionary, we choose the word pair with maximum probability. For the Chinese word c_j, we only choose the English word e_i as its translation which has the maximum probability and vice versa for the reverse direction. Then we combine the lexical dictionary from two directions into one lexical dictionary under current grain.

(2) Och dictionary: we extract phrase pairs consistent with the word alignment of the last grain [13]. The phrase probabilities are calculated by the linear weighting of the two lexical probabilities and two frequency probabilities.

(3) Sentence length information.

3.2. The Step-Extended-Feature Algorithm

In order to obtain sub-sentence alignment efficiently, the features are extended step by step. Suppose we have segmented the sentence pair into sub-sentences sc_1^M and se_1^N, and we choose features $f_1, f_2, \cdots, f_D$ to compute the matching scores between the sub-sentence pairs, the order of the features is arranged beforehand. Because we are concerned with finding the best alignment rather than the exact matching scores, we add a lower-ordered feature to the total matching score only if the higher-ordered features cannot distinguish the alignments by a pre-defined threshold. For each sc_m in sc_1^M we find a se_n from se_1^N to align to sc_m according to our matching score algorithm and we can get a Chinese-to-English sub-sentence alignment. In the same way we can get an English-to-Chinese sub-sentence alignment. Our final sub-sentence alignment is the union of the two sub-sentence alignments.

After we get the alignment matrix of sub-sentence alignment, we merge these sub-sentences to obtain blocks according to the following rule:

$$\{(bc_r) <=> (be_r)\} \in Set\,Of\,Block \Leftrightarrow$$
$$\forall sc_m \in bc_r : (sc_m, se_n) \in S \rightarrow se_n \in be_r$$
$$and\ \forall se_n \in be_r : (sc_m, se_n) \in S \rightarrow sc_m \in bc_r$$

3.3. Combining the Grain-based Word Alignments

In each grain, we train GIZA++ on the blocks to get word alignment of the whole sentence pair. So we have word alignments based on three levels of grain. There are three $I \times J$ word alignment matrixes $A_k = [a_{k_{ij}}]$ and three $I \times J$ probability matrixes $P_k = [p_{k_{ij}}]$. For $A_k = [a_{k_{ij}}]$, we have $a_{k_{ij}} = 1$ if e_i is aligned with c_j and $a_{k_{ij}} = 0$ otherwise. For $P_k = [p_{k_{ij}}]$, we have $p_{k_{ij}} = (p_k(c_j|e_i) + p_k(e_i|c_j))/2$, which means the average probabilities of the bidirectional word dictionary generated by GIZA++ under the k-th grain.

In order to incorporate all the grain-based word alignments, we obtain a new $I \times J$ probability matrix $P = [p_{ij}]$ using a linear weighted model to combine the four matrixes in Equation (4).

$$p_{ij} = \sum_{k=1}^{K} \lambda_k a_{k_{ij}} p_{k_{ij}} \quad (4)$$

Here we set all λ_k, $1 \le k \le K$ as 1. The final $I \times J$ word alignment matrix $A = [a_{ij}]$ is obtained by using the algorithm in Figure 3. Here $SplitScore_i$ is the selected splitting point by using the maximal separation criteria. $u_{p_{ij} \ge p}$ is the mean probability of those probabilities larger than or equal to p.

Input: $I \times J$ probability matrix $P = [p_{ij}]$;
Output: $I \times J$ word alignment matrix $A = [a_{ij}]$;
1. Initialize $A = [a_{ij}] = 0$.
2. For each i do
3. Find $SplitScore_i = \arg\max_{p \in \{p_{ij}, 1 \le j \le J\}} (u_{p_{ij} \ge p} - u_{p_{ij} < p})$;
4. $a_{ij} = 1$ if $p_{ij} > SplitScore_i$;
5. End for;
6. Output $A = [a_{ij}]$;

Figure 3: The algorithm of obtaining final word alignment.

4. EXPERIMENTS

Our experiments are conducted on a parallel corpus of Chinese-English bilingual texts. We use two evaluation metrics to measure the quality of word alignment. One is AER (Alignment Error Rate). The other is to compare the performance of MT system to see how much the BLEU score has been improved. We run MOSES decoder provided in the open source Moses package[2] by the default parameters. We only train 3-gram language model with SRILM on the English side of our training data.

Table 1 gives the detailed statistics of the experimental data. The training data include the training data released by the IWSLT'07 (International Workshop on Spoken Language Translation 2007) and the filtered data from the web resources[3]. We use the test data of IWSLT'07 as our test data of MT system and a test data with human-annotated word-level alignment for computing our AER.

Table 2 shows the AER and BLEU scores under each grain in our WAMG. K is the grains we used. K=1 is our baseline. K=2 is the linear weighted result of the first and second grain. K=3 is the linear weighted result by all the three grains. We achieve an improvement of about 12% relative in AER. Furthermore, our approach outperforms the baseline significantly with a relative improvement of 2.8% in BLEU.

Analysis of our experimental results reveals that the wrong word alignment generated by our approach are due to two factors: the matching score features employed in our model are unable to provide enough information; and the segmentation features are

[2] http://www.statmt.org./moses/.
[3] http://iwslt07.itc.it/ menu/resources.html.

heuristic ones like the punctuation and key words. The two factors neglect some useful information and introduce some noise.

Table 1:Statistics of training corpus and test corpus. Voc. denotes vocabulary; ASL means average sentence length.

Task	Data		Sentence	Voc.	ASL
AER	Train	Chi.	231,518	11,657	7.68
		Eng.	231,518	12,428	8.31
	Test	Chi.	499	1,109	8.86
		Eng.	499	1,069	9.64
BLEU	Train	Chi.	232,017	11,659	7.68
		Eng.	232,017	12,429	8.31
	Test	Chi	489	862	6.47
		Eng	2,934	1,527	7.69

Table 2: Comparison of AER and BLEU for MGWA

Grain	AER[%]	BLEU[%]
K=1	16.14	41.77
K=2	14.89	42.44
K=3	14.21	42.92

5. CONCLUSIONS

In this paper, we propose a novel approach of word alignment based on multi-grain model WAMG) that improves the quality of word alignments and trades off the precision and recall simultaneously by linearly weighting all the word alignment from the different grains. The experimental results have shown that WAMG outperforms the baseline GIZA++ relatively by about 12% in AER and improves the performance of MT system relatively by 2.8% in BLEU.

Our future work will focus on incorporating more features into our model to achieve better performance and measuring the quality of word alignment on a larger data.

6. ACKNOWLEDGEMENTS

The research work has been funded by the Natural Science Foundation of China under Grant No.60575043 and 60736014, the National Key Technology R&D Program under Grant No. 2006BAH03B02, the Hi-Tech Research and Development Program ("863" Program) of China under Grant No. 2006AA01Z194, 2006AA010108-4, and Nokia Research Center, Beijing as well.

7. REFERENCES

[1] Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne and David Talbot, "Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation", in *IWSLT'2005*. Pittsburgh,USA, Oct. 2005.

[2] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer, "The mathematics of statistical machine translation: Parameter estimation", *Computational Linguistics*, vol.19,no.2, pp.263–311, 1993.

[3] Vogel Stephan, Hermann Ney and Christoph Tillman, "HMM-Based Word Alignment in Statistical Translation", in *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, Copenhagen, Denmark, August ,1996, pp. 836-841.

[4] Franz J. Och and Hermann Ney, "A systematic comparison of various statistical alignment models", *Computational Linguistics*, vol.29, no.1, pp.19-51, March, 2003.

[5] Frank Smadja, Kathleen R. McKeown, Vasileios Hatzivassiloglou, "Translating Collocations for Bilingual Lexicons: A Statistical Approach", *Computational Linguistics*, vol.22, no.1, pp. 1-38, 1996.

[6] Sue J. Ker and Jason S. Chang, "A class-based approach to word alignment", *Computational Linguistics*, vol.23, no.2, pp.313-343, June, 1997.

[7] Dan Melamed, "Models of translational equivalence among words", *Computational Linguistics*, vol.26, no.2, pp.221–249, 2000.

[8] Phil Blunsom and Trevor Cohn, "Discriminative Word Alignment with Conditional Random Fields", in *Proceedings of ACL'2006*, Sydney, Australia, 2006.

[9] A. Ittycheriah and S. Roukos, "A maximum entropy word aligner for Arabic-English machine translation", in *Proceedings of HLT-EMNLP'2005* , October, 2005, pp. 89–96.

[10] Yang Liu, Qun Liu, and Shouxun Lin, "Log-linear models for word alignment", in *Proceedings of ACL'2005*, Ann Arbor, 2005, pp. 459–466.

[11] Ben Taskar, Simon Lacoste-Julien, and Dan Klein, "A discriminative matching approach to word alignment", in *Proceedings of EMNLP*, 2005.

[12] Necip F. Ayan, Bonnie J. Dorr, and Christof Monz, "Neuralign: Combining word alignments using neural networks". In *Proceedings of EMNLP'2005*, pp. 65–72, 2005.

[13] Franz Josef Och and Hermann Ney, "The alignment template approach to statistical machine translation", *Computational Linguistics*, vol.30, pp.417–449, 2004.