輪講発表資料 平成 18 年 4 月 14 日

統計的枠組みを統合した木構造アラインメント
A Probabilistic Framework for Structure-based Alignment

黒橋研究室 電子情報学専攻 56430 中澤 敏明

Abstract

In this paper, we propose a new method for alignment using correspondence pattern score (CP-score). This method is a hybrid one combining structural preference, which is a characteristic of EBMT, and statistical formalism, which is a characteristic of SMT. We conduct experiments on travel domain corpus, and achieve an F-measure improvement of 2.6% over a baseline system.

1 Introduction

In machine translation tasks, how to align the training parallel corpus with high accuracy is a big problem, and thus a number of studies have been done. The alignment methods can be categorized into two groups: one is probabilistic methods and the other is heuristic methods with structural information.

Probabilistic methods are mainly used in Statistical Machine Translation (SMT) systems [11]. The main issue is how to decompose the alignment probabilities $Pr(\mathbf{A}|\mathbf{S},\mathbf{T})$ reasonably to make good use of some approximations, where $\mathbf{A}$ represents an alignment, $\mathbf{S}$ represents a source sentence, and $\mathbf{T}$ represents a target sentence.

The simplest statistical method is based on word level alignment, in which the IBM Model [2] is mostly used as the baseline method. Recently, more sophisticated methods have been proposed by [12] and [13], which handle not only a word but a larger block which is usually a multiple word or a phrase. However, even if these methods are oriented to use larger block or structure, data sparseness problem is still a big problem on its way. For this reason, it is not easy to achieve high performance for the language pair whose linguistic structures are quite different from each other.

On the other hand, heuristic structural methods are usually used in Example Based Machine Translation (EBMT) systems. They use heuristic rules in alignment procedure, and can easily use NLP resources, such as a morphological analyzer and a syntactic analyzer, to grasp characteristics of language pairs with large difference in linguistic structure.

[9] proposed a kind of tree structure called “Logical Form”, which is an unordered graph representing the relations among the most meaningful elements of a sentence. With this structure they proposed a “best-first” alignment method. This method starts from the nodes with the tightest lexical correspondence and then goes to close nodes from the first nodes. [5] used parsed tree structure of the original sentence, and then aligned the trees with some heuristic rules which constrain the order of alignment.

Although these structure-based methods utilize profound knowledge of NLP and achieve high accuracy, the manner of alignment is still heuristic, which is often not general-purpose. To resolve this issue, [4] proposed a probabilistic tree-based alignment between Korean and English. They use some cloning operations to calculate the probability, so they make the structure more complicated. Moreover, it is not apparent that the same operations are effective and fit for different language pairs.

In this paper, we introduce a probabilistic framework for structure-based alignment. This is an integration of the SMT’s statistical alignment and EBMT’s deep use of structural preference by decomposing the alignment probability $Pr(\mathbf{A}|\mathbf{S},\mathbf{T})$.

For one of the decompositions, we propose correspondence pattern score (CP-score), which can evaluate the relations between all the pair of correspondences. We define correspondence pattern (CP) as the positional relationship between correspondences, and CP-score as the probability of each CP. Experimental results prove that alignment performance is improved with this new criterion.

In the following section, we briefly introduce the basic structure-based alignment module in our machine translation system. In Section 3, the CP-score as a decomposition of the alignment probability is defined, and the disambiguation method of correspondences using CP-score

is introduced. Moreover, in Section 4, we propose how to use the CP-score in a more precise way by the modification of English dependency structure. In addition, we further decompose the alignment probability into several feature functions and integrate them by means of the maximum entropy method. We performed some experiments to evaluate our proposal, and it is reported in Section 5. At last, we give a short conclusion and introduce our future work.

2 Basic Structure-based Alignment Method

In this section, we briefly introduce our alignment module of a machine translation system. This alignment module is used as a baseline method in the experiments, and from the alignment results, we calculate the CP-score stated in Section 3.2.

Our machine translation system works mainly for Japanese-English, and the alignment is achieved by the following steps, using a Japanese parser, an English parser, and a bilingual dictionary.

1. Dependency analysis of Japanese and English sentences.
2. Detection of word/phrase correspondences.
3. Disambiguation of correspondences.
4. Handling of remaining words.

2.1 Dependency Analysis of Japanese and English Sentences

Japanese sentences are converted into dependency structures using the morphological analyzer, JUMAN [7], and the dependency analyzer, KNP [6]. Japanese dependency structure consists of nodes which correspond with content words. Function words such as postpositions, affixes, and auxiliary verbs are included in content words' nodes.

For English sentences, Charniak's nlparser is used to convert them into phrase structures [3], and then they are transformed into dependency structures by rules defining head words for phrases. In the same way as Japanese, each node in this dependency tree consists of a content word and related function words.

2.2 Detection of Word/Phrase Correspondences

Correspondences between Japanese word/phrase and English word/phrase are detected by a Japanese-English dictionary.

At this moment, the dictionary is not probabilistic. By looking up the whole pair of Japanese words and English words in the dictionary, correspondences are detected deterministically.

In addition to the dictionary, we also handle transliteration. For possible person names and place names suggested by the morphological analyzer and Katakana words (Katakana is a Japanese alphabet usually used for loan words), their possible transliterations are produced and their similarity with words in the English sentence is calculated based on the edit distance. If there are similar word pairs whose edit distance exceeds a threshold, they are handled as a correspondence.

2.3 Disambiguation of Correspondences

There are sometimes two types of correspondence ambiguities. One is that there are the same words in the sentence, the other is that one word has some different meanings.

We resolve these ambiguities with some harmonious criteria. Suppose there is a correspondence X with ambiguity, and there is an unambiguous correspondence Y with the distance n in the Japanese dependency tree and the distance m in the English dependency tree, we give a score $1/n + 1/m$ to the correspondence X. Here we define the distance of correspondences as the number of traversing nodes in a dependency tree.

Then, we hold a assumption that the closer Y is to X, the more strongly Y supports X. Consequently, we accept the ambiguous correspondence with the highest score and reject the others conflicting with the accepted one. This calculation is repeated until all the ambiguous correspondences are resolved.

Figure 1 is an example of correspondence disambiguation. The root of a tree is placed at the extreme left and phrases are placed from top to bottom, and the correspondences of circled words were detected by a bilingual dictionary.

There is only one determined correspondence "日本 (Japan) ↔ Japan" in the example. Considering this correspondence as a clue, the scores are calculated, and the

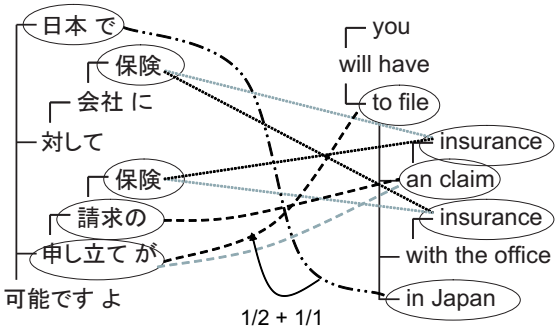

Figure 1: Example of disambiguation.

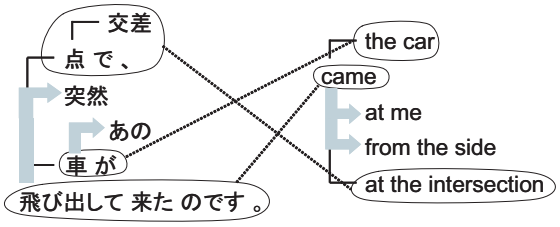

Figure 2: Example of extend.

correspondence “申し立て (allegation) ↔ file” with the highest score is adopted. At the same time, the conflicting correspondence “申し立て (allegation) ↔ claim” is rejected. After that, the correspondence “請求 (claim) ↔ claim” is unambiguous so that it is adopted.

This strategy is effective but heuristic, and what is worse, sometimes the same scores are given to ambiguous correspondences. In this case, the ambiguities cannot be resolved.

2.4 Handling of Remaining Words

The alignment procedure so far found some correspondences in parallel sentences. Then, we merge the remaining nodes into existing correspondences.

First, the root nodes of the dependency trees are handled as follows. In the given training data, we suppose that all parallel sentences have appropriate translation relation. Accordingly, if neither of the root nodes (of the Japanese dependency tree and the English dependency tree) is included in any correspondences, the new correspondence between the two root nodes is generated. If either root node is remaining, it is merged into the correspondence of the other root node.

Then, for both Japanese remaining node and English remaining node, if it is within a base NP and another node in the NP is in a correspondence, it is merged into the correspondence. At last, other remaining nodes are merged into correspondences of their parent (or ancestor) nodes.

For example in Figure 2, “あの (that)” is merged into the correspondence “車 (car) ↔ the car”, since it is within an NP. “突然 (suddenly)”, “at me” and “from the side” are merged into their parent correspondence, “飛び出して来たのです (rush out) ↔ came”.

3 Statistical Structure-based Alignment

Our previous alignment module uses some heuristic rules so that it cannot handle all the ambiguous correspondences. To solve this problem, we introduce a probabilistic framework which combines the statistical information from the whole parallel corpus into our previous alignment module. To do this, it is necessary to decompose the alignment probability $Pr(\mathbf{A}|\mathbf{S},\mathbf{T})$ in a reasonable way. For one of the decomposed components, we propose correspondence pattern score (CP-score).

3.1 Correspondence Pattern (CP)

CP is defined as a positional pattern of a pair of correspondences based on the tree structure. Figure 3 is an abstract example of dependency tree structure. Circles represent nodes (they are actually phrases) and the numbers represent the correspondences: same numbered nodes are correspondent with each other, and blank nodes have no correspondence (means aligned to NULL).

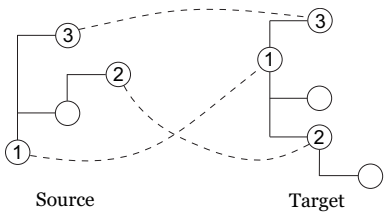

Figure 3: Example of dependency tree structure.

The nodes connected by broken lines represent correspondences between source and target language (Figure 3). CP is defined as a relation between two correspondences. In Figure 3, there are three correspondences (1,

2 and 3), and so there are three combinations of correspondences (1-2, 1-3 and 2-3) as shown in Figure 4. CP will be extracted from each combinations in the following steps.

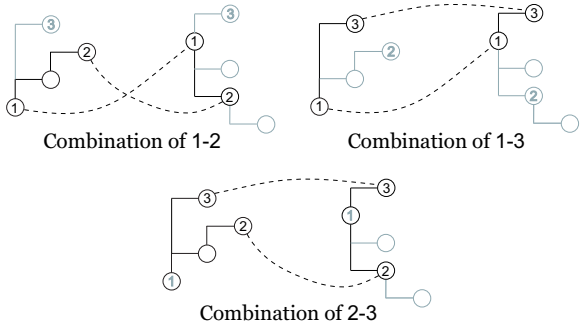

Figure 4: Combinations of correspondences.

As for the source language tree structure, there are two kinds of relation between aligned nodes: lineal relation and collateral relation (Figure 5). In Figure 5, X and Y nodes are the focused aligned nodes and the gradated nodes are the common parent node from each aligned node.

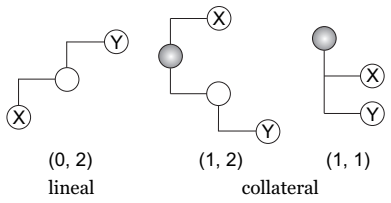

Figure 5: Two types of relation.

All the relations are distinguished by the distance: for lineal relations, we use distance between two nodes, and for collateral relations, we use two distances to the common parent node. For example, the relations in Figure 5 can be represented with two numerals: (0, 2) for the first, (1, 2) for the second, and (1, 1) for the last. The numerals are written in ascending order because what is important is the pair of numerals, and we treat (0, 2) and (2, 0) as the same relation.

It is the same for the target language, but there is one thing that we have to care, which is that the order of numerals for target language should correspond to that of source language. Then, CP is represented as a combination of the two relations of focused correspondences between source and target language, which means that CPs are distinguished by four numerals. Figure 6 shows some CPs extracted from Figure 5. The CP for the first example is (0, 2, 1, 2) and for the second is (0, 2, 2, 1). Note that the two CPs are different from each other and the score is also different even though the tree structures are the same. This difference can be held by the constraint of numeral order for target language.

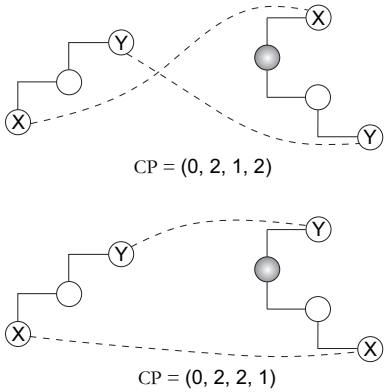

Figure 6: Examples of CP.

3.2 Calculation of CP-score

We assign a score to each CP, which is calculated by counting the frequency of each CP from the aligned parallel corpus by the method introduced in the previous section, and then divide them by the total frequency of all CPs.

We calculate the CP-score by the aligned parallel corpus which is made by our original alignment module. The corpus is a travel domain corpus (BTEC) used for a training data set in IWSLT2005[1], which contains 20k Japanese-English parallel sentences[2].

3.3 Alignment Disambiguation with CP-score

Ambiguous correspondences can be resolved with CP-score. In case there is one or more ambiguous correspondences, we generate all the possible alignment candidates. And then, the alignment score for each of the possible alignments is calculated by means of CP-score. Finally, we adopt the best alignment which gets the highest score.

We define the alignment score (AS) as the product of

[1] http://www.is.cs.cmu.edu/iwslt2005/

[2] It is worth trying to calculate the CP-score in an unsupervised way like the EM algorithm. According to our preliminary experiment, it is not very effective to use iterative estimation for CP-score.

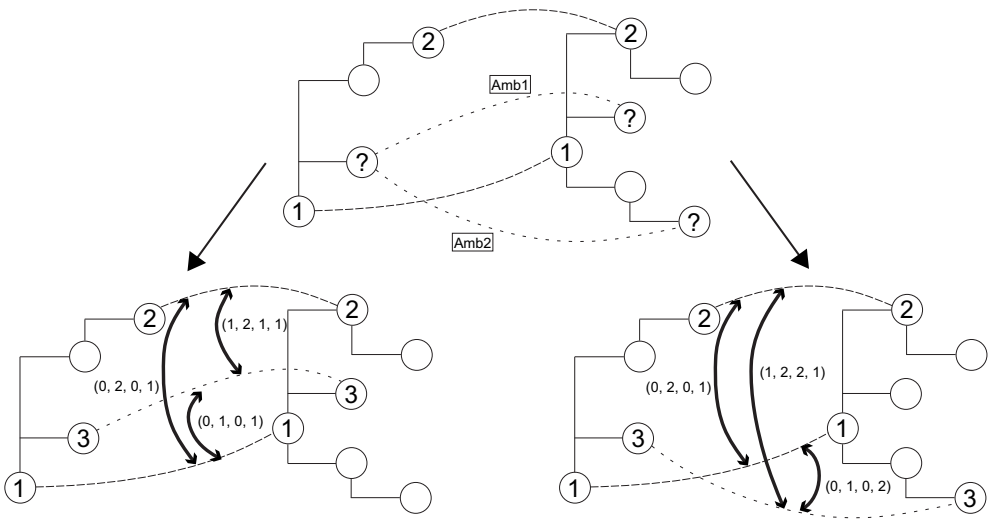

Figure 7: Example of disambiguation.

CP-scores between all pairs of correspondences:

$$AS = \prod_{i=1}^{M-1} \prod_{j=i+1}^{M} CP\text{-}score_{i,j} \quad (1)$$

where M is the number of correspondences in the sentence pair.

Figure 7 is an example of disambiguation. There are two ambiguous correspondences, and from that, two possible alignments are generated. At last, we adopt the alignment with higher AS.

4 Consideration and Optimization of Dependency Structure Possibilities

4.1 Modification of Dependency Structure

Automatic syntactic analysis inevitably contains some errors, such as word segmentation errors and parsing errors. These errors may decrease the accuracy of structure-based alignment.

As for English, parsing is particularly difficult, because of the following characteristics: one word can have various parts-of-speech and meanings, and there is PP-attachment ambiguity. These problems cause parse errors, and sometimes make the alignment accuracy worse.

On the other hand, the head of a Japanese sentence is always put at the end of the sentence. Therefore, it is easier to parse Japanese sentences and to get correct dependency trees. Consequently, we give great confidence on the Japanese dependency trees, and try to modify only the English dependency trees in this paper. This is performed by applying n-best dependency structures to our alignment module. After the alignment of each dependency structure, the AS can be calculated as a product of all the CP-scores as given in Equation (1). Then, we choose the alignment with the highest AS, and the best dependency structure is acquired simultaneously.

4.2 Combination of Various Scores Using Maximum Entropy Method

Although we consider only AS so far, the score of each English structure is also useful. The problem arising here is that we do not know how to combine the CP-score and the structure score. Taking this problem into consideration, parameters that weigh the various scores appropriately are necessary. To find the parameters, we employ the maximum entropy (ME) method [1].

Alignment methods with ME are reported in [10] and [8]. Following these methods, we define alignment probability $Pr(\mathbf{A}|\mathbf{S},\mathbf{T})$ as follows:

$$Pr(\mathbf{A}|\mathbf{S},\mathbf{T}) = \frac{\exp\left[\sum_{m=1}^{M} \lambda_m h_m(\mathbf{A},\mathbf{S},\mathbf{T})\right]}{\sum_{\mathbf{A}'} \exp\left[\sum_{m=1}^{M} \lambda_m h_m(\mathbf{A}',\mathbf{S},\mathbf{T})\right]}$$

where $h_m(\mathbf{A},\mathbf{S},\mathbf{T})$ are feature functions, and λ_m are model parameters.

In addition to the AS and the parse score, we introduce some other feature functions which are thought to be effective.

AS: The AS of the sentence.

English parse score: The Charniak's parse score output itself.

Depth pattern score (DP-score): The score of the pattern of depth from head node of the sentence. Depth pattern (DP) is represented with four numerals, and DP-score is calculated in the same way of CP-score.

Probability of the lexicon: The product of all the probabilities of the lexicon which is included in each correspondence. For lexical probability, we use the output of the word alignment tool GIZA++ by [11]. In case of multiple word correspondences in one phrase correspondence, the one which has the largest probability is chosen as a representation. We use both source-to-target and target-to-source alignment probabilities independently.

Coverage of the correspondences: The ratio of aligned nodes compared to the total number of nodes. This is calculated in both source and target sentence independently.

Average size of the correspondences: Our alignment method can handle not only one-to-one node alignment but one-to-many or many-to-many nodes alignment. Furthermore, variations of dependency structure occasionally merge some phrases into one phrase. For these reasons, the average size of the correspondence can affect the alignment. This is calculated in both source and target sentence independently.

Training data set for ME is automatically generated from the Charniak's n-best output. Moreover, if there are some ambiguous correspondences, we can also generate alignment candidates in the way shown in section 3.3. All of them are compared to the gold-standard alignment, which is created in Section 5, then, the one best candidate based on F-measure is marked correct and the others marked incorrect.

5 Experimental Results and Discussion

5.1 Experiments and Results

We selected 500 moderately long sentences from the BTEC corpus of IWSLT2005 training data set and manually annotated phrase-to-phrase alignment to them. This is because our alignment module is based on tree structure composed of phrases. A part of the criteria for annotation are listed below:

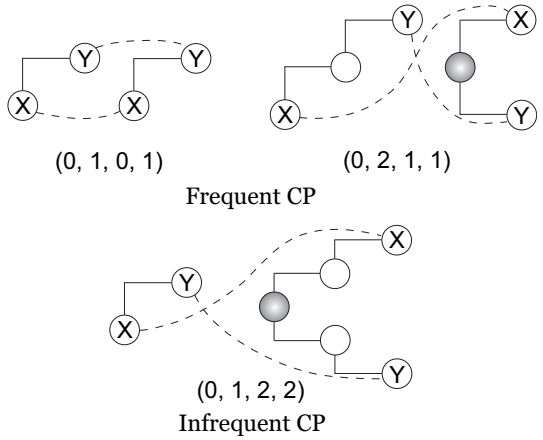

Figure 8: Examples of frequent and infrequent CP.

- Merge function words into content words.
- Remove all punctuation marks.
- Omitted subjects or objects are merged into head verb phrase.

We conducted 5-fold cross validation, in which 400 sentences are used for ME training and 100 sentences for testing. CP-score and DP-score are calculated from the base alignment results of the IWSLT2005 training corpus. Figure 8 shows examples of frequent and infrequent CPs, and we can see that CPs whose relation of source language is similar to that of target language occur more frequently.

The evaluation results are shown in Table 1. "Baseline" is the heuristic method introduced in Section 2. "CP-score" is the disambiguation of correspondence with CP-score introduced in section 3. "ES Modification", which is an abbreviation of English Structure Modification, utilizes n-best parse results. "ME" is the integration method of nine feature functions by ME. For "ES Modification" and "ME", we use 3-best Charniak's parse results.

There are three evaluations. The first one is the result of whole 500 sentences and evaluated with all the function words. "CP-score" performed better than the "Baseline". "ES Modification" was a little worse than "CP-score" and only the result of "ME" is the statistically significant compared to that of the baseline method ($p < 0.05$).

The second evaluation was done also on 500 sentences, but all the function words are eliminated for the appropriate comparison with the statistical alignment method, GIZA++. Function words make the alignment accuracy of GIZA++ worse because GIZA++ is a word-based

Table 1: Evaluation results.

Method	All sentences w/ function words	All sentences w/o function words	Ambiguous sentences w/ function words
Baseline	63.86	65.14	60.43
+ CP-score	64.21	65.54	61.60
+ ES Modification	64.20	65.47	61.61
+ ME	64.58	66.03	63.00
GIZA	22.14	52.85	23.76

Table 2: Effect of the number of parse candidates.

	ES Modification	ME
3-best	64.20	64.58
5-best	63.99	64.47
10-best	63.94	64.38

alignment tool and our gold-standard data is based on phrases.

GIZA++ was trained using whole 20k sentences of IWSLT2005 training corpus, in which function words are left, and all the words are lowercased and converted to base forms. The alignment was performed in one direction (Japanese to English). Because we have an advantage of using the dictionary, we can see that our method achieved much higher accuracy.

In the third evaluation, we selected 142 sentences out of 500, which have ambiguous correspondences, and all the function words are included. From this evaluation, we can see the improvement measurably: 1.2% improvement for "CP-score", and 2.6% for "ME" over the "Baseline". The improvements of all of our methods are statistically significant ($p < 0.05$).

5.2 Discussion

We discuss the following five points derived form the experimental results.

Disambiguation with CP-score: If two correspondences are in the same clause of the source sentence, they are mostly in the same clause of the target sentence. We do not distinguish a main clause and a subordinate clause on the tree structure now. Therefore, even if there is an ambiguous correspondence which crosses the clause, we do not impose any penalties. We need to resolve this problem because disambiguation errors cause further serious alignment errors.

ES Modification: The accuracy of "ES Modification" was worse than that of "CP-score". This is because we do not care the Charniak's parse scores, and CP-score sometimes tends to choose incorrect parse results. However, this problem was solved by the ME integration. Table 2 shows the effect of the number of parse candidates, which are the Charniak's n-best. As the number of candidates increases, the accuracy falls off, but the degree of the decrease of "ME" is smaller than that of "ES Modification".

Sentence complexity: One of the reasons of small improvement of the results comes from the complexity of the sentences. Our proposed method works effectively for long and complex sentences in particular. The BTEC corpus' sentences are short, and come from spoken language, and so it is difficult to find correspondences using a dictionary. We will test our method in other corpus such as newspaper and patent in the future.

Preciseness of dictionary: The dictionary we used is not precise nor concise. If there is an erroneous correspondence by the dictionary, it makes bad effects on alignment. On the other hand, if the number of entries is small, it becomes hard to align accurately. Preparing a precise and concise dictionary is also our future work.

Parse error of Japanese: Not only English parse results but Japanese parse results are sometimes wrong, and this leads to alignment errors. The Japanese dependency analyzer KNP can produce n-best parse results, and so in the future, we will consider parse results of both Japanese and English.

6 Conclusion

We have proposed a probabilistic framework to improve structure-based alignment. As one of the decompositions of alignment probability $Pr(\mathbf{A}|\mathbf{S}, \mathbf{T})$, we pro-

posed new criteria CP-score for evaluating alignment. With the CP-score we succeed to utilized probabilities for structure-based alignment and achieved higher alignment accuracy.

We also integrated the ME model into our alignment approach. We utilized nine feature functions, which made it possible to modify the dependency tree structure and lead to much higher alignment accuracy.

What we need to do in the future is to sophisticate the CP and CP-score, one way is to consider the clause, and to select the feature functions to improve our ME model, and moreover, we will test our method on other corpora which consist of long sentences.

Publications

- IJCNLP 2005
- IWSLT 2005
- COLING-ACL 2006 (Under audition)

References

[1] Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Association for Computational Linguistics*, 19(2):263–312, 1993.

[3] Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan, June 2005.

[4] Daniel Gildea. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 80–87, 2003.

[5] Declan Groves, Mary Hearne, and Andy Way. Robust sub-sentential alignment of phrase-structure trees. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 1072–1078, 2004.

[6] Sadao Kurohashi and Makoto Nagao. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4):507–534, 1994.

[7] Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of The International Workshop on Sharable Natural Language*, pages 22–28, 1994.

[8] Yang Liu, Qun Liu, and Shouxun Lin. Log-linear models for word alignment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 459–466, 2005.

[9] Arul Menezes and Stephen D. Richardson. A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL) Workshop on Data-Driven Machine Translation*, pages 39–46, 2001.

[10] Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, 2002.

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

[12] Taro Watanabe, Kenji Imamura, and Eiichiro Sumita. Statistical machine translation based on hierarchical phrase alignment. In *9th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 188–197, 2002.

[13] Ying Zhang and Stephan Vogel. An efficient phrase-to-phrase alignment model for arbitrarily long phrase and long corpora. In *European Association for Machine Translation 2005 Conference Proceedings*, pages 294–301, 2005.