Volume 4, Issue 9, September 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Hybrid Approach to Improve the Performance of Word Alignment in English-Hindi Bilingual Corpora with Scarce Resources

Jyoti Srivastava *
Indian Institute of Information Technology,
Allahabad, India

Sudip Sanyal
Indian Institute of Information Technology,
Allahabad, India

Abstract—This paper presents a hybrid approach to improve the performance of word alignment for English-Hindi language pair with scarce resources. Sentence segmentation and part of speech tag information is used to improve the performance of word alignment. IBM Model 1 is used as word alignment algorithm. This paper demonstrates increase of precision, recall and F-measure by approximately 12%, 11%, 12% respectively and reduction in Alignment Error Rate (AER) by approximately 12%. Experiments of this paper are based on TDIL corpus of size 1000 where 5% of the corpus is used for testing and rest corpus is used for training.

Keywords—Clause Identification, Natural Language Processing, Part of Speech Tagger, Statistical Machine Translation, Word alignment

I. INTRODUCTION

Word alignment is the task of identifying the correct translation relationships among the words of a bilingual corpus [1] [2]. It is the first step in Statistical machine translation. This paper focused on the word alignment of English-Hindi language pair where the resources are scarce. The performance of the word alignment algorithm is increased when the size of the parallel corpus is increased. But to find a huge parallel corpus is very rare and expensive; so this approach is an effort to deal with small size of parallel corpus.

Many word alignment techniques have been developed so far in Natural Language Processing (NLP). But, word alignment techniques between English and Hindi did not have much progress due to two main reasons: complex structure of the participating languages and scarcity of Hindi-language resources. Longer sentence pair in bilingual corpus consumes much memory and CPU time in training and also decreases the performance of word alignment. This paper presents a simple method to improve the word-alignment accuracy of IBM Models in which these limitations have been overcome. This paper not only breaks the longer sentences into shorter ones, even it breaks all the sentences into shorter segments so that word alignment process becomes much easier. This task is performed by using clause identifier and part of speech (POS) tagger prior to word alignment algorithm.

The basic hypothesis of this work is that if we break the sentences into shorter segments prior to the calculation of the various probabilities then the performance of word alignment will get improved. Here the word segments refer to the group of words which are most probable for alignment with each other. The words in this group need not to be consecutive i.e. the order of the words in this method have no importance as we are using IBM Model 1 which works on bag of words paradigm. To break the sentences into shorter segments, a clause identifier algorithm and POS tagger is used [3] [4]. The word alignment model used here is the IBM Model 1.

IBM Model 1 is a word alignment model which is widely used for working with parallel bilingual corpora [1]. IBM Model 1 was originally developed for providing reasonable parameter estimates to initialize more complex word-alignment models. There are also other word alignment models like IBM Models 2-5 and HMM. IBM Models 1-2 have the problem of fertility and distortion which is solved in IBM Models 3-5. All the IBM models are relevant, because the training (using Expectation Maximization algorithm) starts with the simplest IBM Model 1 for a few iterations and then proceeds through iterations of the more complex models all the way to IBM Model 5. So we can expect that if IBM Model 1 will be improved, the higher IBM Models will also get improved. Thus, we will work primarily with IBM Model 1 and expect that any improvement in this model will also get reflected in the higher IBM models. This approach not only decreases the alignment error rate (AER) of word alignment but also speeds up the process of word alignment as demonstrated by the examples discussed in section 3.

The rest of the paper is structured as follows. Section 2 briefly discusses word alignment and some previous works in which some statistical and rule based strategy is used for word alignment. Section 3 describes the method proposed to improve the performance of word alignment by using a hybrid method. Section 4 presents the data and evaluation metrics used to evaluate the method. Section 5 presents the result of the method and an analysis of the same. Section 6 finishes the paper with some conclusions and proposal for future work.

II. RELATED WORK

Word alignment is useful in many applications in the area of NLP. It is an essential step of SMT [1] [2]. Word aligned corpus is used to extract multiword expressions with semantic meaning [5]. Word alignment is also useful in automatic extraction of bilingual lexicon and terminology [6]. It is also used to transfer language tools developed for one language to other languages. Many NLP applications are enhanced and can improve their performance by using word alignment of better-quality [7]. So better word alignment is a central issue to achieve good performance in many NLP applications.

Many word alignment techniques are proposed in the literature. However, words may be poorly aligned in long sentence pairs in practice, which will then degrade the performance of word alignment. On the other hand, training a system using long sentence pairs usually cost much more memory and CPU time. In order to make good use of the information carried by long sentence pairs, it is necessary to segment long sentences into shorter ones. Xu et al. took the first step in this problem by introducing a method of performing sentence segmentation based on modified IBM Translation Model 1 [8]. To our knowledge, little further research has been done in this area. By splitting the long sentence pairs into shorter ones in the training corpus, we can get better alignment quality [8] [9].

In Systran, as described by Hutchins and Somers, conjunct and relative clauses were segmented in a preprocessing step [10]. Chandrasekar applied a sentence simplification method to machine translation, where sentences are split at conjunctions, relative pronouns, etc. before translation [11]. Rao et al. describe a clause-wise translation strategy within an English-Hindi transfer-based MT system [12]. Kim and Ehara proposed a rule-based method for splitting long Japanese sentences for Japanese-to- English translation [13]. Marcu provides cue phrases for English in his thesis which can be used to break the sentences into clauses [14].

Sudoh et al. proposed a method to perform clause level alignment of the parallel corpus and to translate clauses (all clauses identified by a syntactic parser) as a unit to improve long-distance reordering in a specialized domain, English-Japanese translation of research paper abstracts in the medical domain [15]. They applied automatic clause identification in both training and testing time but only at source side not on the target side. They segment the sentences into clauses at target side by using source clauses and word alignment from source language to target language.

Ramanathan et al. used clause based constraints at the time of testing to improve the performance of SMT [16]. They used clause identification on both source and target but only during testing and not at the time of training. Srivastava and Sanyal also used automatic clause segmentation method to improve the performance of the word alignment [3].

An approach which converted word alignment task to the problem of integer linear programming is proposed by Bodrumlu et al. [17]. Probabilistic generative approaches like IBM Model 1-5, HMM and LEAF are based on hidden alignment variable and they finally optimize word maps using Expectation Maximization (EM) algorithm [1] [18] [19]. Most practitioners still use IBM Models and HMM models for word alignment. The tool based on these methods is GIZA++. HMM word alignment models and GIZA++ which is an implementation of the IBM Model 1-5 are the most widely-used alignment system [1].

In particular, for English-Hindi word alignment, some work has been reported. Chatterjee and Agrawal have conducted experiments on manually lemmatized parallel corpus using recency vector based approach [20]. A hybrid approach based on local word grouping, cognates, nearest aligned neighbor, dictionary lookup, transliteration similarity and finally language dependent grammar rules for the alignment of English-Hindi bilingual corpus is given by Aswani and Gaizauskas [21]. Hindi is a partial free word order language. In a Hindi sentence, order of word groups is not fixed, but order of words within a group is fixed [22]. Venkataramani and Gupta provide a corpus-augmented method of word alignment for English-Hindi with scarce resources [23]; they used two existing word alignment tools: GIZA++ and NATools. Word alignment algorithm performs better on POS tagged parallel corpus [4]. Several recent works incorporate syntactic features into alignment to improve the performance of word alignment. Thurmair discuss about strength and weaknesses of rule based MT and statistical MT and then decided to use a hybrid system which take advantage of both the system; rule based and statistical system for machine translation [24].

Research in NLP for Hindi and the other Indian languages is still in its beginning. Reliable linguistic resources like bilingual parallel corpus, lemmatizer, stemmer, etc. are not easily available. Several educational institutions are working on different projects of these resources. Hence, applying the above methods on large corpus is not easy and thus provides a strong motivation for experimenting with techniques that perform well with small corpus.

Based on the above literature, we decided to make a hybrid model for word alignment in which statistical model is the base model and some rule based strategy like POS tagger and clause identification are applied to improve the performance of the word alignment task. The proposed method is not limited to a specific clause identification method or any specific POS tagger; any method can be employed, if their clause definition matches the proposed method where clauses are independently translated and POS tagger is of high accuracy. Our approach is related to sentence simplification and is intended to obtain simple and short source and target segments for better word alignment.

III. EXPERIMENTAL METHODOLOGY

In order to verify the contribution of POS tagger and Clause identification in improvement of the performance of word alignment, we propose the following steps. First POS (Part-of-Speech) tagger is applied. POS tagged corpus is used as input for the clause identification method. Then we apply clause identification on this corpus to split the longer sentences into shorter ones. The conventional IBM Model 1 is then used to compute the alignment probability for each source word of the source sentence with each target word of the target sentence in the POS tagged clause separated parallel corpus. By applying POS tagger and clause identification on the parallel corpus the computation is reduced drastically and the performance is improved as explained in section 3.3.

The approach proposed in this paper computes the alignment probability of a source word of a source sentence with only those target words of the target sentence which have the same POS tag as the source word and belong to the corresponding target clause.

An example sentence pair from TDIL corpus is given below to demonstrate the benefit of the proposed method for word alignment:

Sentence pair: 14(English Sentence length) * 17(Hindi Sentence length) = 238
English: Jaipur, popularly known as the Pink City, is the capital of Rajasthan state, India. → 14
Hindi: जयपुर जो गुलाबी नगर के नाम से जाना जाता है, भारत के राजस्थान राज्य की राजधानी है। → 17

Here the English sentence has 14 words and the Hindi sentence has 17 words. When we will apply word alignment algorithm IBM Model 1 on this sentence pair then the algorithm will calculate the probability of each of the 14 English word with each of the 17 Hindi word. So there will be total 238 (14*17) combinations of source and target words for which the probability will be calculated in one iteration. Moreover, more than one iteration is required to maximize the probability and get the optimized word alignment. So we can see that the number of computations is large.

A. Contribution of Clause identification in word alignment:

Problematic long sentences often include embedded clauses such as relative clauses. Such an embedded (subordinate) clause can usually be translated almost independently of words outside the clause. Longer sentences consume more time in training of word alignment and also degrade the performance.

Now we will demonstrate the benefit of using clause pair instead of sentence pair for word alignment:

We break the sentence pair given above into clause pair as given below:

Clause pair 1: 7*10 = 70
[Jaipur, popularly known as the Pink City] → 7
[जयपुर जो गुलाबी नगर के नाम से जाना जाता है] → 10

Clause pair 2: 7*7 = 49
[,is the capital of Rajasthan state, India.] → 7
[,भारत के राजस्थान राज्य की राजधानी है।] → 7

There are 2 clause pairs in this sentence pair, the cue phrase used to split the sentence is ", (comma)". Clause length is given in front of each clause and the total number of computation needed to calculate the word alignment probability is given in front of each clause pair. Thus the number of computation for probability calculation will be reduced to 119 (70+49) which is really small in comparison to 238.

B. Contribution of POS tag in word alignment:

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

POS tags also solve the problem of word sense disambiguates at a small scale. For example: "book" will have different translations depending on whether it is used as noun or verb as given in the following example.

Read the **book (noun)**.
Book (verb) the ticket.

In a purely statistical technique like IBM Models, "book" will be translated to "book" either as verb (बुक करना) or as noun (किताब) but not both. On the other hand, if the sentence containing the word "book" is POS tagged, then "book" will be tagged as verb or noun, depending on the sentence. Now, while looking for translations of the word "book", the translation system will search for their corresponding translation with tag information using statistical techniques and it will find the right translation. Thus this approach also deals with the problem of word sense disambiguation at certain level.

POS tagger decreases the computation and time requirement by decreasing the combination of source and target words to calculate the probability. Now we will demonstrate the benefit of using POS tagger on the above sentence pair for word alignment:

POS tagged Sentence pair:

English: Jaipur/NN ,/SYM popularly/RB known/VB as/PSP the/DT Pink/NN City/NN ,/SYM is/VB the/DT capital/NN of/PSP Rajasthan/NN state/NN ,/SYM India/NN ./SYM

Hindi: जयपुर/NN जो/PRP गुलाबी/NN नगर/NN के/PSP नाम/NN से/PSP जाना/VB जाता/VB है/VB ,/SYM भारत/NN के/PSP राजस्थान/NN राज्य/NN की/PSP राजधानी/NN है/VB ।/SYM

After breaking this POS tagged sentence pair into same tag word pair:

{(DT) (the) (NULL)}

{(NN) (Jaipur Pink City capital Rajasthan state India) (जयपुर गुलाबी नगर नाम भारत राजस्थान राज्य राजधानी)}

{(PRP) (NULL) (जो)}

{(PSP) (as, of) (के से की)}

{(VB) (known, is) (जाना जाता है)}
{(RB) (popularly) (NULL)}

{(SYM) (, .) (, ।)}

There are 7 tag pair in this sentence pair. Thus the number of computation for probability calculation will be reduced upto 75 (1+56+1+6+6+1+4) which is really small in comparison to 238.

As we can see in the above example that by breaking the sentence pair into clauses, the number of computation for probability calculation is reduced from 238 to 119 and by applying POS tagger on this sentence pair, the number of computation for probability calculation is reduced from 238 to 75. So both approaches decrease the number of computation to calculate the probability for word alignment thus saving time and energy.

C. Contribution of combination of Clause identification and POS tag in word alignment:

Now we will demonstrate the benefit of using a combination of both approaches discussed above for the same example sentence pair given above for word alignment. When the sentence pair is broken into clause pair and the POS tagger is applied then the resulting tag pairs will be:

For Clause pair 1:

English: Jaipur/NN ,/SYM popularly/RB known/VB as/PSP the/DT Pink/NN City/NN
Hindi: जयपुर/NN जो/PRP गुलाबी/NN नगर/NN के/PSP नाम/NN से/PSP जाना/VB जाता/VB है/VB

After breaking this POS tagged clause pair into same tag word pair:
{(DT) (the) (NULL)}
{(NN) (Jaipur Pink City) (जयपुर गुलाबी नगर नाम)}
{(RB) (popularly) (NULL)}
{(PRP) (NULL) (जो)}
{(VB) (known) (जाना जाता है)}
{(PSP) (as) (के से)}
{(SYM) (,) (NULL)}

The number of computation to calculate probability for clause pair 1 after Applying POS tagger, is 21 (1+12+1+1+3+2+1).

For Clause pair 2:

English: ,/SYM is/VB the/DT capital/NN of/PSP Rajasthan/NN state/NN ,/SYM India/NN ./SYM
Hindi: ,/SYM भारत/NN के/PSP राजस्थान/NN राज्य/NN की/PSP राजधानी/NN है/VB ।/SYM

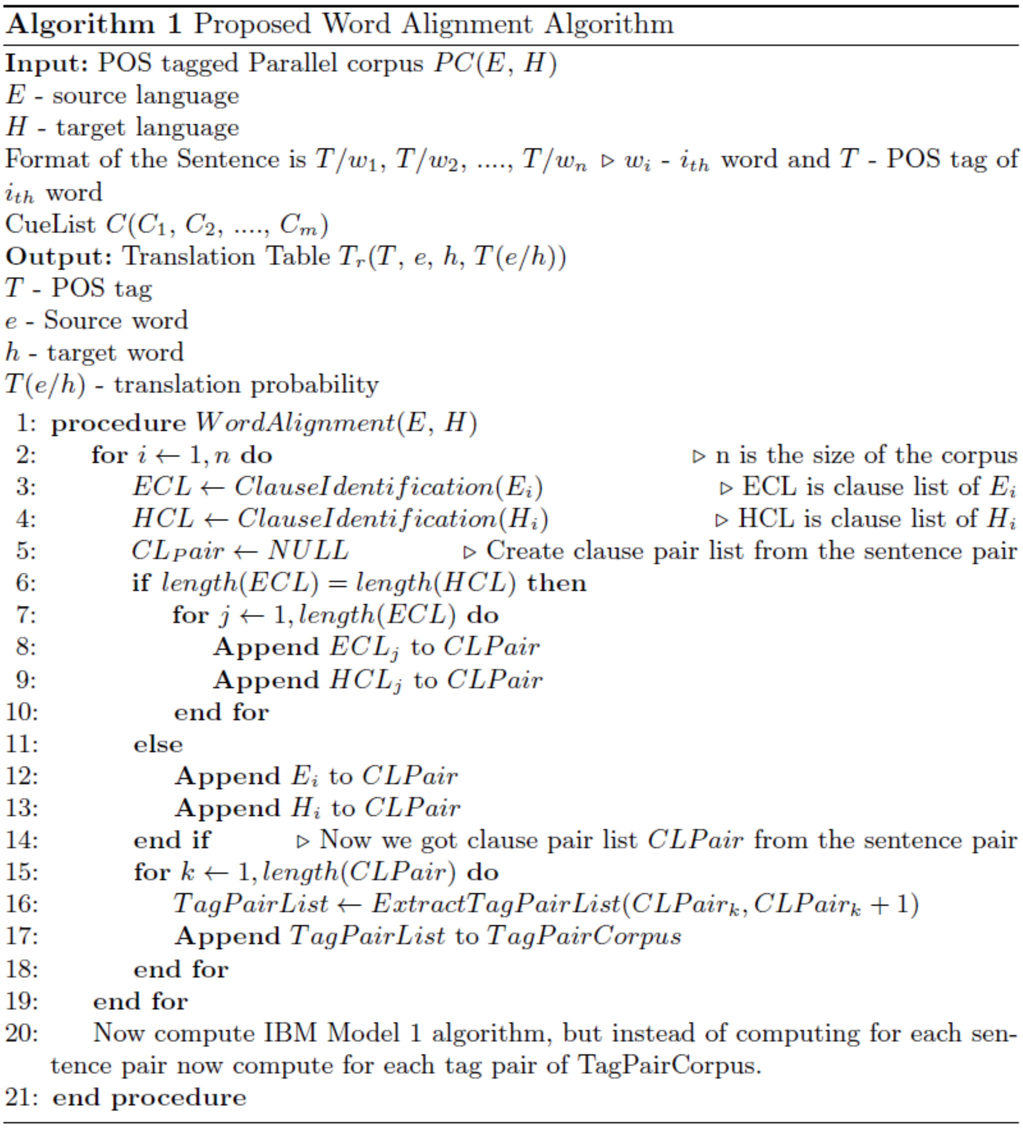
After breaking this POS tagged clause pair into same tag word pair:
{(DT) (the) (NULL)}
{(NN) (capital Rajasthan state India) (भारत राजस्थान राज्य राजधानी)}
{(PSP) (of) (के की)}
{(VB) (is) (है)}
{(SYM) (, .) (, ।)}

The number of computation to calculate probability for clause pair 2 after Applying POS tagger, is 24 (1+16+2+1+4). Thus number of computation to calculate the probability for the complete sentence pair is reduced up to 45(21+24) which is really very small in comparison to 238. Thus using the combination of clause identification and POS tagger, we get better performance in word alignment in comparison to applying both of them individually.

Proposed approach is given in algorithm 1. Step 3 and step 4 of the algorithm 1 used automatic clause identification method proposed by Srivastava and Sanyal [3]. Step 16 of the algorithm is talking about extraction of words having same POS tag which is described by Srivastava and Sanyal [4].

Apart from the reduction in complexity we also observe that the above steps will improve the accuracy of the probability calculations. This happens because the number of available target words is reduced when we calculate the probability with each source word with its POS tag information. This can be seen from the following example.

When we calculate the probability on plain corpus then the probability of word "Jaipur" with "जयपुर" is 0.903 and probability of word "capital" with "राजधानी" is 0.776. While when the probability is calculated on the clause-separated POS tagged corpus, the probability of word "Jaipur" with "जयपुर" is increased up to 0.999 and probability of word "capital" with "राजधानी" is increased up to 0.999. So when we use clause separated and POS tagged corpus instead of plain corpus for word alignment the computation time is reduced as well as the performance gets improved.



Algorithm 1 Proposed Word Alignment Algorithm

```
Input: POS tagged Parallel corpus $PC(E, H)$
$E$ - source language
$H$ - target language
Format of the Sentence is $T/w_1$, $T/w_2$, ...., $T/w_n$ ▷ $w_i$ - $i_{th}$ word and $T$ - POS tag of $i_{th}$ word
CueList $C(C_1, C_2, ...., C_m)$
Output: Translation Table $T_r(T, e, h, T(e/h))$
$T$ - POS tag
$e$ - Source word
$h$ - target word
$T(e/h)$ - translation probability
procedure $WordAlignment(E, H)$
    for $i \leftarrow 1, n$ do                                   ▷ n is the size of the corpus
        $ECL \leftarrow ClauseIdentification(E_i)$               ▷ ECL is clause list of $E_i$
        $HCL \leftarrow ClauseIdentification(H_i)$               ▷ HCL is clause list of $H_i$
        $CLpair \leftarrow NULL$               ▷ Create clause pair list from the sentence pair
        if $length(ECL) = length(HCL)$ then
            for $j \leftarrow 1, length(ECL)$ do
                Append $ECL_j$ to $CLPair$
                Append $HCL_j$ to $CLPair$
            end for
        else
            Append $E_i$ to $CLPair$
            Append $H_i$ to $CLPair$
        end if          ▷ Now we got clause pair list $CLPair$ from the sentence pair
        for $k \leftarrow 1, length(CLPair)$ do
            $TagPairList \leftarrow ExtractTagPairList(CLPair_k, CLPair_k + 1)$
            Append $TagPairList$ to $TagPairCorpus$
        end for
    end for
    Now compute IBM Model 1 algorithm, but instead of computing for each sen-
  tence pair now compute for each tag pair of TagPairCorpus.
end procedure
```

IV. Data and Evaluation

TDIL parallel corpus with 1000 sentence pairs is used for experimentation. The system was trained on 950 sentences of TDIL corpus (English-Hindi). The remaining 50 sentences (5% of the corpus) were used for testing. The performance of the system was measured in terms of precision, recall and F-measure which were also frequently used in the previous word alignment literature [25]. Och and Ney defined a fourth measure which is alignment error rate (AER) [7]. AER is a measure of quality of word alignment.

Alignment A is the set of alignments produced by the alignment model under testing. With a gold standard alignment *G*, each such alignment set consisting of two sets A_S , A_P and, G_S , G_P corresponding to Sure (*S*) and Probable (*P*) alignments, these performance statistics are defined as

$$P_T = \frac{|A_T \cap G_T|}{|A_T|} \qquad (1)$$

$$R_T = \frac{|A_T \cap G_T|}{|G_T|} \qquad (2)$$

$$F_T = \frac{2P_T R_T}{P_T + R_T} \qquad (3)$$

$$AER = 1 - \frac{|A_P \cap G_S| + |A_P \cap G_P|}{|A_P| + |G_S|} \qquad (4)$$

Where *T* is the alignment type, and can be set to either *S* or *P*.

V. Result and Discussion

To evaluate the effectiveness of using POS tagger and automatic clause segmentation with word alignment model IBM Model 1, a comparison is performed between the proposed method and IBM Model 1 without any preprocessing. We have tested our approach and the conventional IBM Models 1 for different corpus size from 200 to 950 and the results are given in Fig. 1 and 2.

Table 1 shows the result of word alignment on plain (original) corpus, automatic clause separated corpus, POS tagged corpus and automatic clause separated POS tagged corpus for 200 sentences.

Table I: Effectiveness Of Using POS Tagger, Clause Identification, And Combination Of Both On Word Alignment Model IBM Model 1

Corpus size	**Methodology**	**Precision (%)**	**Recall (%)**	**F-Measure (%)**	**AER (%)**
200	IBM Model 1	34.10	41.37	37.39	62.61
200	IBM Model 1 + Clause Identification	42.71	47.59	45.02	54.98
200	IBM Model 1+ POS tagger	48.48	54.70	51.40	48.60
200	IBM Model 1+Clause Identification + POS tagger	50.00	55.46	52.59	47.41

Here we can see the effectiveness of using POS tagger and Clause identification individually with IBM Model 1. But when we use the combination of both, the result is improved very much. As you can see the AER on plain corpus for 200 sentences is 62.61% while on clause separated POS tagged corpus of 200 sentences, AER is 47.41%. So the difference between AER on plain corpus and automatic clause separated POS tagged corpus is approximately 15% which is really very large. These results are computed for corpus of 200 sentences. Similarly for other corpus size, we get improved performance for word alignment.

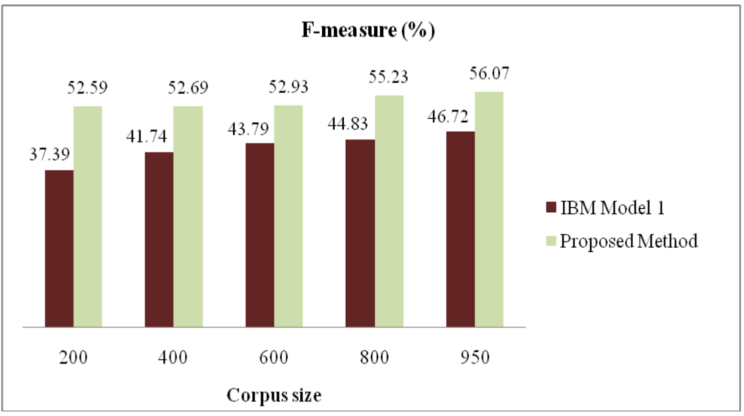


Fig. 1 Comparison of F-meaure computed on IBM Model 1 with plain corpus and automatic clause separated POS tagged corpus

Figure 1 and Figure 2 demonstrate the effect of POS tagger and automatic clause separation on the IBM Model 1 as the corpus size increases. IBM Model 1 algorithm used here is described by Koehn [26].

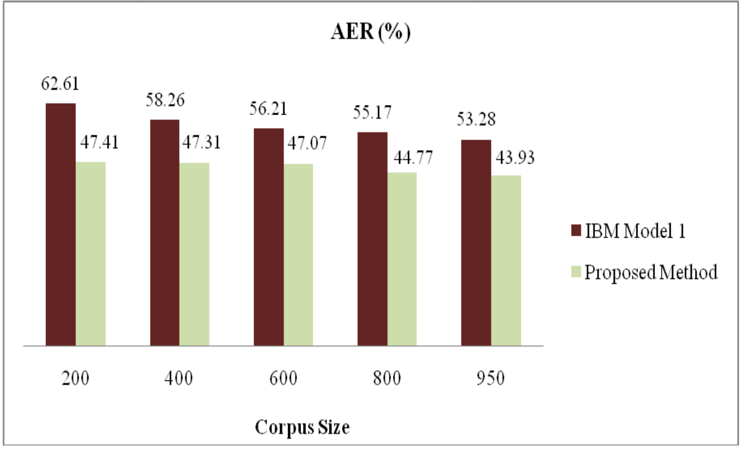


Fig. 2 Comparison of AER computed on IBM Model 1 with plain corpus and automatic clause separated POS tagged corpus

These results demonstrate that when the size of the parallel corpus is increased, F-measure increases and AER (Alignment Error Rate) decreases. Obviously, it is expected and is true for any statistical algorithm that when the sample size increases, the percentage error decreases. But after some limit the increment in the sample size do not cause decrement in percentage error. So we want to decrease this limit by using some rule based strategy with the statistical algorithm. The results in Fig. 1 and Fig. 2 demonstrate that by breaking the parallel corpus into clauses and applying POS tag, we can improve the performance of the word alignment.

The other practical benefit of using the proposed approach is that when we use shorter segments of the sentence rather than complete sentence for training, the training time is reduced since MT training tends to run faster on shorter sentences.

VI. CONCLUSION AND FUTURE WORK

This paper presented a hybrid approach of word alignment in English-Hindi when the resources are scarce. It is demonstrated that it is possible to improve the performance of IBM Model 1 in terms of F-measure and AER by about 12%, simply by breaking sentences into clauses and using POS tagger. This paper focused on using the short segments of the sentences for training of word alignment model. All the conducted experiments provide evidence that using clause separated POS tagged corpus with any IBM Model performs better when compared to the use of plain corpus in IBM Model 1-5, for the task of word alignment. This experiment provides new way to extend this approach for other Indian languages. This paper focuses on developing suitable word alignment schemes in parallel texts where the size of the corpus is not too large. The scarcity of the resources suggests that purely statistical techniques are not suitable for the task.

Although these word alignment results are encouraging, we can further improve it by providing the solutions to the problems discussed in section 5.2. The second problems discussed in section 5.2 can be solved by introducing a morphological analyzer in statistical model of word alignment. By solving these problems and using higher IBM Models which also deals with fertility and distortion, we expect further improvements in the performance of word alignment.

ACKNOWLEDGMENTS

We are thankful to IIIT Allahabad for providing the suitable infrastructure for research. This research has been funded by Tata Consultancy Services (TCS). We are really thankful to Indian Language Technology Proliferation and Deployment Centre Team for providing Sample tourism parallel corpus. We are grateful to the organizers of ACL 2005 workshop to make the word alignment evaluation code as open source.

REFERENCES

[1] Brown PF, Della Pietra SA, Della Pietra VJ and Mercer RL (1993) The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2), pp. 263–311.

[2] Gale WA and Church KW. (1991) Identifying word correspondences in parallel texts. Fourth DARPA Workshop on Speech and Natural Language. Asilomar, pp. 152– 157.

[3] Jyoti Srivastava and Sudip Sanyal "Segmenting Long Sentence Pairs to Improve Word Alignment in English-Hindi Parallel Corpora", 8th International conference on Natural Language Processing, Kanazawa, Japan. Published in Advances in Natural Language Processing, Lecture Notes in Computer Science Volume 7614, October 2012, pp 97-107.

[4] Jyoti Srivastava and Sudip Sanyal "A Hybrid Approach for Word Alignment in English-Hindi Parallel Corpora with Scarce Resources", International Conference on Asian Language Processing (IALP 2012), November 2012, Hanoi, Vietnam, pp 185-188.

[5] Caseli, H. D., Ramisch, C., Nunes, M. D. V. and Villavicencio, A., 2010. Alignment-based extraction of multiword expressions. *Language Resources and Evaluation*, 44 (1-2), pp. 59-77.

[6] Smadja F A., McKeown KR. and Hatzivassiloglou V, Translating Collocations for Bilingual Lexicons: A Statistical Approach. Computational Linguistics, 1996, 22(1), pp. 1-38.

[7] Och FJ and Ney H (2003) A systematic comparison of various statistical alignment models. In Computational Linguistics, 29(1), pp. 19–51.

[8] J. Xu, R. Zens, , and H. Ney, "Sentence segmentation using IBM word alignment model 1," in Proc. the 10th Annual Conference of the European Association for Machine Translation, Budapest, Hungary, May 2005, pp. 280–287.

[9] Meng, B., Huang, S., Dai, X., Chen, J.: Segmenting long sentence pairs for statistical machine translation. In: International Conference on Asian Language Processing, Singapore (Dec 7-9 2009)

[10] Hutchins, J., and Somers, H., An Introduction to Machine Translation, pages 175–189, Academic Press, 1992.

[11] Chandrasekar, R., A Hybrid Approach to Machine Translation using Man Machine Communication, Ph.D. thesis, Tata Institute of Fundamental Research, Mumbai, 1994.

[12] Rao, D., Mohanraj, K., Hegde, J., Mehta, V., and Mahadane, P., A practical framework for syntactic transfer of compound-complex sentences for English-Hindi machine translation, Proceedings of KBCS, 2000.

[13] Kim, Y.-B., Ehara, T.: A method for partitioning of long Japanese sentences with subject resolution in J/E machine translation. In: Proc. International Conference on Computer Processing of Oriental Languages, pp. 467–473 (1994)

[14] Marcu, D., The Rhetorical Parsing, Summarization and Generation of Natural Language Texts, Ph.D. thesis, Department of Computer Science, University of Toronto, Toronto, Canada, December 1997.

[15] Sudoh, K., Duh, K., Tsukada, H., Hirao, T., and Nagata, M., Divide and translate: improving long distance reordering in statistical machine translation, Workshop on Statistical Machine Translation and Metrics, 2010.

[16] Ramanathan, A., Bhattacharyya, P., Visweswariah, K., Ladha, K., and Gandhe, A., Clause-Based Reordering Constraints to Improve Statistical Machine Translation. Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, November, 2011 pp. 1351-1355

[17] Bodrumlu T, Knight K and Ravi S (2009) A New Objective Function for Word Alignment. NAACL Workshop on Integer Linear Programming for NLP. Boulder, Colorado, pp. 28–35.

[18] Vogel S, Ney H and Tillmann C (1996) HMM-based word alignment in statistical translation. ACL, Volume 2, pp. 836-841.

[19] Fraser A and Marcu D (2007) Getting the structure right for word alignment: LEAF. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, pp. 51–60.

[20] Chatterjee N and Agrawal S. (2006) Word Alignment in English Hindi Parallel Corpus Using Recency-Vector Approach: Some Studies. 21st International Conference on Computational Linguistics, Sydney, Australia, pp. 17-21.

[21] Aswani N and Gaizauskas R. (2005a) A hybrid approach to align sentences and words in English-Hindi parallel corpora. In Proceedings of the ACL Workshop on Building and Using Parallel Texts, Ann Arbor, pp. 57–64.

[22] Ray PR, V. H, Sarkar S and Basu (2003) A Part of Speech Tagging and Local Word Grouping Techniques for Natural Language Parsing in Hindi. 1st International Conference on Natural Language Processing (ICON 2003); Mysore.

[23] Venkataramani E and Gupta D. (2010) English-Hindi Automatic Word Alignment with Scarce Resources. International Conference on Asian Language Processing, IEEE. pp. 253-256.

[24] Thurmair G, 2005: Hybrid Architectures for Machine Translation Systems. in: Language Resources and Evaluation 39, 1, 91-108.

[25] Patrik Lambert, Adrià de Gispert, Rafael Banchs and José B. Mariño. 2005. Guidelines for Word Alignment Evaluation and Manual Alignment. Language Resources and Evaluation, 39 (4) pp. 267-285. Springer.

[26] Koehn P (2010) Statistical Machine Translation. Book. Cambridge University Press, Published in the United States of America by Cambridge University Press, New York.