Word Alignment in English-Hindi Parallel Corpus Using Recency-Vector Approach: Some Studies

Niladri Chatterjee
Department of Mathematics
Indian Institute of Technology Delhi
Hauz Khas, New Delhi
INDIA - 110016
niladri_iitd@yahoo.com

Saumya Agrawal
Department of Mathematics
Indian Institute of Technology
Kharagpur, West Bengal
INDIA - 721302
saumya_agrawal2000@yahoo.co.in

Abstract

Word alignment using recency-vector based approach has recently become popular. One major advantage of these techniques is that unlike other approaches they perform well even if the size of the parallel corpora is small. This makes these algorithms worth-studying for languages where resources are scarce. In this work we studied the performance of two very popular recency-vector based approaches, proposed in (Fung and McKeown, 1994) and (Somers, 1998), respectively, for word alignment in English-Hindi parallel corpus. But performance of the above algorithms was not found to be satisfactory. However, subsequent addition of some new constraints improved the performance of the recency-vector based alignment technique significantly for the said corpus. The present paper discusses the new version of the algorithm and its performance in detail.

1 Introduction

Several approaches including statistical techniques (Gale and Church, 1991; Brown et al., 1993), lexical techniques (Huang and Choi, 2000; Tiedemann, 2003) and hybrid techniques (Ahrenberg et al., 2000), have been pursued to design schemes for word alignment which aims at establishing links between words of a source language and a target language in a parallel corpus. All these schemes rely heavily on rich linguistic resources, either in the form of huge data of parallel texts or various language/grammar related tools, such as parser, tagger, morphological analyser etc.

Recency vector based approach has been proposed as an alternative strategy for word alignment. Approaches based on recency vectors typically consider the positions of the word in the corresponding texts rather than sentence boundaries. Two algorithms of this type can be found in (Fung and McKeown, 1994) and (Somers, 1998). The algorithms first compute the *position vector* V_w for the word w in the text. Typically, V_w is of the form $\langle p_1 p_2 \ldots p_k \rangle$, where the p_is indicate the positions of the word w in a text T. A new vector R_w, called the *recency vector*, is computed using the position vector V_w, and is defined as $\langle p_2 - p_1, p_3 - p_2, \ldots, p_k - p_{k-1} \rangle$. In order to compute the alignment of a given word in the source language text, the recency vector of the word is compared with the recency vector of each target language word and the similarity between them is measured by computing a matching cost associated with the recency vectors using dynamic programming. The target language word having the least cost is selected as the aligned word.

The results given in the above references show that the algorithms worked quite well in aligning words in parallel corpora of language pairs consisting of various European languages and Chinese, Japanese, taken pair-wise. Precision of about 70% could be achieved using these algorithms. The major advantage of this approach is that it can work even on a relatively small dataset and it does not rely on rich language resources.

The above advantage motivated us to study the effectiveness of these algorithms for aligning words in English-Hindi parallel texts. The corpus used for this work is described in Table 1. It has been made manually from three different sources: children's storybooks, English to Hindi translation book material, and advertisements. We shall call

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 649–656,
Sydney, July 2006. ©2006 Association for Computational Linguistics

the three corpora as Storybook corpus, Sentence corpus and Advertisement corpus, respectively.

2 Word Alignment Algorithm: Recency Vector Based Approach

DK-vec algorithm given in (Fung and McKeown, 1994) uses the following dynamic programming based approach to compute the matching cost $C(n, m)$ of two vectors v_1 and v_2 of lengths n and m, respectively. The cost is calculated recursively using the following formula,

$$\begin{aligned} C(i,j) &= |(v_1(i) - v_2(j)| + \min\{C(i-1,j), \\ &\quad C(i-1,j-1), C(i,j-1)\} \end{aligned}$$

where i and j have values from 2 to n and 2 to m respectively, n and m being the number of distinct words in source and target language corpus respectively. Note that $v_l(k)$ denotes the k^{th} entry of the vector v_l, for $l = 1$ and 2. The costs are initialised as follows.

$$\begin{aligned} C(1,1) &= |v_1(1) - v_2(1)|; \\ C(i,1) &= |v_1(i) - v_2(1)| + C(i-1,1); \\ C(1,j) &= |v_1(1) - v_2(j)| + C(1,j-1); \end{aligned}$$

The word in the target language that has the minimum normalized cost $(C(n, m)/(n + m))$ is taken as the translation of the word considered in the source text.

One major shortcoming of the above scheme is its high computational complexity i.e. $O(mn)$. A variation of the above scheme has been proposed in (Somers, 1998) which has a much lower computational complexity $O(\min(m, n))$. In this new scheme, a distance called Levenshtein distance(S) is successively measured using :

$$\begin{aligned} S = S + \min\{&|v_1(i+1) - v_2(j)|, \\ &|v_1(i+1) - v_2(j+1)|, |v_1(i) - v_2(j+1)|\} \end{aligned}$$

The word in the target text having the minimum value of S (Levenshtein difference) is considered to be the translation of the word in the source text.

2.1 Constraints Used in the Dynamic Programming Algorithms

In order to reduce the complexity of the dynamic programming algorithm certain constraints have been proposed in (Fung and McKeown, 1994).

1. *Starting Point Constraint*: The constraint imposed is: |first-occurrence of source language word (w_1) - first-occurrence of target language word w_2| $< \frac{1}{2}*$(length of the text).

2. *Euclidean distance constraint*: The constraint imposed is: $\sqrt{(m_1 - m_2)^2 + (s_1 - s_2)^2} < T$, where m_j and s_j are the mean and standard deviation, respectively, of the recency vector of w_j, $j =$ 1 or 2. Here, T is some predefined threshold:

3. *Length Constraint*: The constraint imposed is: $\frac{1}{2} * f_2 < f_1 < 2 * f_2$, where f_1 and f_2 are the frequencies of occurrence of w_1 and w_2, in their respective texts.

2.2 Experiments with DK-vec Algorithm

The results of the application of this algorithm have been very poor when applied on the three English to Hindi parallel corpora mentioned above without imposing any constraints.

We then experimented by varying the values of the parameters in the constraints in order to observe their effects on the accuracy of alignment. As was suggested in (Somers, 1998), we also observed that the Euclidean distance constraint is not very beneficial when the corpus size is small. So this constraint has not been considered in our subsequent experiments. Starting point constraint imposes a range within which the search for the matching word is restricted. Although Fung and McKeown suggested the range to be half of the length of the text, we felt that the optimum value of this range will vary from text to text depending on the type of corpus, length ratio of the two texts etc. Table 2 shows the results obtained on applying the DK vec algorithm on Sentence corpus for different lower values of range. Similar results were obtained for the other two corpora. The maximum increase observed in the F-score is around 0.062 for the Sentence corpus, 0.03 for the Story book corpus and 0.05 for the Advertisement corpus. None of these improvements can be considered to be significant.

2.3 Experiments with Somers' Algorithm

The algorithm provided by Somers works by first finding all the minimum score word pairs using dynamic programming, and then applying three filters *Multiple Alignment Selection filter*, *Best Alignment Score Selection* filter and *Frequency Range* constraint to the raw results to increase the accuracy of alignment.

The *Multiple Alignment Selection*(MAS) filter takes care of situations where a single target language word is aligned with the number of source

Corpora	English corpus		Hindi corpus	
	Total words	Distinct words	Total words	Distinct words
Storybook corpus	6545	1079	7381	1587
Sentence corpus	8541	1186	9070	1461
Advertisement corpus	3709	1307	4009	1410

Table 1: Details of English-Hindi Parallel Corpora

Range	Available	Proposed	Correct	P%	R%	F-score
50	516	430	34	7.91	6.59	0.077
150	516	481	51	10.60	09.88	0.102
250	516	506	98	19.37	18.99	0.192
500	516	514	100	19.46	19.38	0.194
700	516	515	94	18.25	18.22	0.182
800	516	515	108	20.97	20.93	0.209
900	516	515	88	17.09	17.05	0.171
1000	516	516	100	19.38	19.38	0.194
2000	516	516	81	15.70	15.70	0.157
4535	516	516	76	14.73	14.73	0.147

Table 2: Results of DK-vec Algorithm on Sentence Corpus for different range

language words. Somers has suggested that in such cases only the word pair that has the minimum alignment score should be considered. Table 3 provides results (see column F-score old) when the raw output is passed through the MAS filters for the three corpora. Note that for all the three corpora a variety of frequency ranges have been considered, and we have observed that the results obtained are slightly better when the MAS filter has been used.

The best F-score is obtained when frequency range is high i.e. 100-150, 100-200. But here the words are very few in number and are primarily pronoun, determiner or conjunction which are not significant from alignment perspective. Also, it was observed that when medium frequency ranges, such as 30-50, are used the best result, in terms of precision, is around 20-28% for the three corpora. However, since the corpus size is small, here too the available and proposed aligned word pairs are very few (below 25). Lower frequency ranges (viz. 2-20 and its sub-ranges) result in the highest number of aligned pairs. We noticd that these aligned word pairs are typically verb, adjective, noun and adverb. But here too the performance of the algorithm may be considered to be unsatisfactory. Although Somers has recommended words in the frequency ranges 10-30 to be considered for alignment, we have considered lower frequency words too in our experiments. This is because the corpus size being small we would otherwise have effectively overlooked many small-frequency words (e.g. noun, verb, adjective) that are significant from the alignment point of view.

Somers has further observed that if the Best Alignment Score Selection (BASS) filter is applied to yield the first few best results of alignment the overall quality of the result improves. Figure 1 shows the results of the experiments done for different alignment score cut-off without considering the Frequency Range constraint on the three corpora. However, it was observed that the performance of the algorithm reduced slightly on introducing this BASS filter.

The above experiments suggest that the performance of the two algorithms is rather poor in the context of English-Hindi parallel texts as compared to other language pairs as shown by Fung and Somers. In the following section we discuss the reasons for the low recall and precision values.

2.4 Why Recall and Precision are Low

We observed that the primary reason for the poor performance of the above algorithms in English - Hindi context is the presence of multiple Hindi equivalents for the same English word. This can happen primarily due to three reasons:

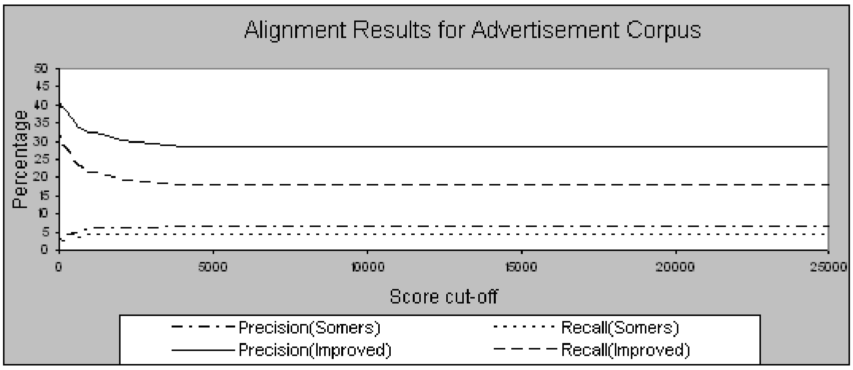

Figure 1: Results of Somers' Algorithm and Improved approach for different score cut-off

Declension of Adjective: Declensions of adjectives are not present in English grammar. No morphological variation in adjectives takes place along with the number and gender of the noun. But, in Hindi, adjectives may have such declensions. For example, the Hindi for "*black*" is *kaalaa* when the noun is masculine singular number (e.g. *black horse* ∼ *kaalaa ghodaa*). But the Hindi translation of "*black horses*" is *kaale ghode*; whereas "*black mare*" is translated as *kaalii ghodii*. Thus the same English word "*black*" may have three Hindi equivalents *kaalaa*, *kaalii*, and *kale* which are to be used judiciously by considering the number and gender of the noun concerned.

Declensions of Pronouns and Nouns: Nouns or pronouns may also have different declensions depending upon the case endings and/or the gender and number of the object. For example, the same English word "*my*" may have different forms (e.g. *meraa*, *merii*, *mere*) when translated in Hindi. For illustration, while "*my book*" is translated as ∼ *merii kitaab*, the translation of "*my name*" is *meraa naam*. This happens because *naam* is masculine in Hindi, while *kitaab* is feminine. (Note that in Hindi there is no concept of Neuter gender). Similar declensions may be found with respect to nouns too. For example, the Hindi equivalent of the word "*hour*" is *ghantaa*. In plural form it becomes *ghante* (e.g. "*two hours*" ∼ *do ghante*). But when used in a prepositional phrase, it becomes *ghanto*. Thus the Hindi translation for "*in two hours*" is *do ghanto mein*.

Verb Morphology: Morphology of verbs in Hindi depends upon the gender, number and person of the subject. There are 11 possible suffixes (e.g *taa*, *te*, *tii*, *egaa*) in Hindi that may be attached to the root Verb to render morphological variations. For illustration,

I read. → *main padtaa hoon* (Masculine) but
main *padtii hoon* (Feminine)
You read. → *tum padte ho* (Masculine) or
tum padtii ho (Feminine)
He will read. → *wah padegaa*.

Due to the presence of multiple Hindi equivalents, the frequencies of word occurrences differ significantly, and thereby jeopardize the calculations. As a consequence, many English words are wrongly aligned.

In the following section we describe certain measures that we propose for improving the efficiency of the recency vector based algorithms for word alignment in English - Hindi parallel texts.

3 Improvements in Word Alignment

In order to take care of morphological variations, we propose to use root words instead of various declensions of the word. For the present work this has been done manually for Hindi. However, algorithms similar to Porter's algorithm may be developed for Hindi too for cleaning a Hindi text of morphological inflections (Ramanathan and Rao, 2003). The modified text, for both English and Hindi, are then subjected to word alignment.

Table 4 gives the details about the root word corpus used to improve the result of word alignment. Here the total number of words for the three types of corpora is greater than the total number of words in the original corpus (Table 1). This is because of the presence of words like "I'll" in the English corpus which have been taken as "I shall" in the root word corpus. Also words like *Unkaa* have been taken as *Un kaa* in the Hindi root word corpus, leading to an increase in the corpus size.

Since we observed (see Section 2.2) that Euclidean distance constraint does not add significantly to the performance, we propose not to use this constraint for English-Hindi word alignment. However, we propose to impose both *frequency range constraint* and *length constraint* (see Section 2.1 and Section 2.3). Instead of the starting point constraint, we have introduced a new constraint, viz. *segment constraint*, to localise the search for the matching words. The starting point constraint expresses range in terms of number of words. However, it has been observed (see section 2.2) that the optimum value of the range varies with the nature of text. Hence no value for range may be identified that applies uniformly on different corpora. Also for noisy corpora the segment constraint is expected to yield better results as the search here is localised better. The proposed segment constraint expresses range in terms of segments. In order to impose this constraint, first the parallel texts are aligned at sentence level. The search for a target language word is then restricted to few segments above and below the current one.

Use of sententially aligned corpora for word alignment has already been recommended in (Brown et al., 1993). However, the requirement there is quite stringent – all the sentences are to be correctly aligned. The segment constraint proposed herein works well even if the text alignment is not perfect. Use of roughly aligned corpora has also been proposed in (Dagan and Gale, 1993) for word alignment in bilingual corpora, where statistical techniques have been used as the underlying alignment scheme. In this work, the sentence level alignment algorithm given in (Gale and Church, 1991) has been used for applying segment constraint. As shown in Table 5, the alignment obtained using this algorithm is not very good (only 70% precision for Storybook corpus). The three aligned root word corpora are then subjected to segment constraint in our experiments.

Next important decision we need to take which dynamic programming algorithm should be used. Results shown in Section 2.2 and 2.3 demonstrate that the performance of DK-vec algorithm and Somers' algorithm are almost at par. Hence keeping in view the improved computational complexity, we choose to use Levenshtein distance as used in Somers' algorithm for comparing recency vectors. In the following subsection we discuss the experimental results of the proposed approach.

3.1 Experimental Results and Comparison with Existing algorithms

We have conducted experiments to determine the number of segments above and below the current segment that should be considered for searching the match of a word for each corpus. In this respect we define *i-segment constraint* in which the search is restricted to the segments $k - i$ to $k + i$ of the target language corpus when the word under consideration is in the segment k of the source language corpus.

Evidently, the value of i depends on the accuracy of sentence alignment. Table 5 suggests that the quality of alignment is different for the three corpora that we considered. Due to the very high precision and recall for Sentence corpus we have restricted our search to the k^{th} segment only, i.e. the value of i is 0. However, since the results are not so good for the Storybook and Advertisement corpora we found after experimenting that the best results were obtained when i was 1. During the experiments it was observed that as the number of segments was lowered or increased from the optimum segment the accuracy of alignment decreased continuously by around 10% for low frequency ranges for the three corpora and remained almost same for high frequency ranges.

Table 3 shows the results obtained when segment constraint is applied on the three corpora at optimum segment range for various frequency ranges. A comparison between the F-score given by algorithm in (Somers, 1998) (the column F-score old in the table) and the F-score obtained by applying the improved scheme (the column F-score new in the table) indicate that the results have improved significantly for low frequency ranges.

It is observed that the accuracy of alignment for almost 95% of the available words has increased significantly. This accounts for words within low frequency range of 2–40 for Sentence corpus, 2–30 for Storybook corpus, and 2–20 for Advertisement corpus. Also, most of the correct word pairs given by the modified approach are verbs, adjectives or nouns. Also it was observed that as the noise in the corpus increased the results became poorer. This accounts for the lowest F-score values for advertisement corpus. The Sentence corpus, however, has been found to be the least noisy, and highest precision and recall values were obtained with this corpus.

Using Somers' second filter on each corpus for the optimum segment we found that the results at low scores were better as shown in Figure 1. The word pairs obtained after applying the modified approach can be used as anchor points for further alignment as well as for vocabulary extraction. In case of the Sentence corpus, best result for anchor points for further alignment lies at the score cut off 1000 where precision and recall are 86.88% and 80.35% respectively. Hence F-score is 0.835 which is very high as compared to 0.173 obtained by Somers' approach and indicates an improvement of 382.65%. Also, here the number of correct word pairs is 198, whereas the algorithms in (Fung and McKeown, 1994) and (Somers, 1998) gave only 62 and 61 correct word pairs, respectively. Hence the results are very useful for vocabulary extraction as well. Similarly, Figure 2 and Figure 3 show significant improvements for the other two corpora. At any score cut-off, the modified approach gives better results than the algorithms proposed in (Somers, 1998).

4 Conclusion

This paper focuses on developing suitable word alignment schemes in parallel texts where the size of the corpus is not large. In languages, where rich linguistic tools are yet to be developed, or available freely, such an algorithm may prove to be beneficial for various NLP activities, such as, vocabulary extraction, alignment etc. This work considers word alignment in English - Hindi parallel corpus, where the size of the corpus used is about 18 thousand words for English and 20 thousand words for Hindi.

The paucity of the resources suggests that statistical techniques are not suitable for the task. On the other hand, Lexicon-based approaches are highly resource-dependent. As a consequence, they could not be considered as suitable schemes. Recency vector based approaches provide a suitable alternative. Variations of this approach have already been used for word alignment in parallel texts involving European languages and Chinese, Japanese. However, our initial experiments with these algorithms on English-Hindi did not produce good results. In order to improve their performances certain measures have been taken. The proposed algorithm improved the performance manifold. This approach can be used for word alignment in language pairs like English-Hindi.

Since the available corpus size is rather small we could not compare the results obtained with various other word alignment algorithms proposed in the literature. In particular we like to compare the proposed scheme with the famous IBM models. We hope that with a much larger corpus size we shall be able to make the necessary comparisons in near future.

References

L. Ahrenberg, M. Merkel, A. Sagvall Hein, and J.Tiedemann. 2000. Evaluation of word alignment systems. In *Proc. 2nd International conference on Linguistic resources and Evaluation (LREC-2000)*, volume 3, pages 1255–1261, Athens, Greece.

P. Brown, S. A. Della Pietra, V. J. Della Pietra, , and R. L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

K. W. Church Dagan, I. and W. A. Gale. 1993. Robust bilingual word alignment for machine aided translation. In *Proc. Workshop on Very Large Corpora: Academic and Industrial Perspectives*, pages 1–8, Columbus, Ohio.

P. Fung and K. McKeown. 1994. Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping. In *Technology Partnerships for Crossing the Language Barrier: Proc. First conference of the Association for Machine Translation in the Americas*, pages 81–88, Columbia, Maryland.

W. A. Gale and K. W. Church. 1991. Identifying word correspondences in parallel texts. In *Proc. Fourth DARPA Workshop on Speech and Natural Language*, pages 152–157. Morgan Kaufmann Publishers, Inc.

Jin-Xia Huang and Key-Sun Choi. 2000. Chinese korean word alignment based on linguistic comparison. In *Proc. 38th annual meeting of the association of computational linguistic*, pages 392–399, Hong Kong.

Ananthakrishnan Ramanathan and Durgesh D. Rao. 2003. A lightweight stemmer for hindi. In *Proc. Workshop of Computational Linguistics for South Asian Languages -Expanding Synergies with Europe, EACL-2003*, pages 42–48, Budapest, Hungary.

H Somers. 1998. Further experiments in bilingual text alignment. *International Journal of Corpus Linguistics*, 3:115–150.

Jörg Tiedemann. 2003. Combining clues word alignment. In *Proc. 10th Conference of The European Chapter of the Association for Computational Linguistics*, pages 339–346, Budapest, Hungary.

Segment Constraint: 0-segment (Sentence Corpus)								
Frequency	a	p	c	P%	R%	F-score	F-score	%
range						(new)	(old)	increase
2-5	285	181	141	77.90	49.74	0.61	0.118	416.90
3-5	147	108	81	75.00	55.10	0.64	0.169	278.69
3-10	211	152	119	78.29	56.40	0.61	0.168	263.10
5-20	146	103	79	76.70	54.12	0.64	0.216	196.29
10-20	49	35	29	82.86	59.18	0.69	0.233	196.14
20-30	19	12	9	75.00	47.37	0.58	0.270	114.62
30-50	14	8	6	75.00	42.86	0.55	0.229	140.17
40-50	4	2	2	100.00	50.00	0.67	0.222	201.80
50-100	15	12	8	66.67	53.33	0.59	0.392	50.51
100-200	6	5	5	100.00	83.33	0.91	0.91	-
200-300	3	3	3	100.00	100.00	1.00	1.00	-
Segment Constraint: 1-segment (Story book Corpus)								
2-5	281	184	89	48.37	31.67	0.38	0.039	874.35
3-5	143	108	52	48.15	36.36	0.41	0.042	876.19
5-10	125	89	35	39.39	28.00	0.33	0.090	266.67
10-20	75	50	25	50.00	33.33	0.40	0.115	247.83
10-30	117	76	39	51.32	33.33	0.41	0.114	259.65
20-30	32	23	11	47.83	34.38	0.37	0.041	802.43
30-40	14	8	2	25.00	14.29	0.18	0.100	80
40-50	7	7	2	28.57	28.57	0.29	0.200	45.00
50-100	11	10	2	20.00	18.18	0.19	0.110	72.72
100-200	5	5	2	40.00	40.00	**0.40**	**0.444**	-
Segment Constraint: 1-segment (Advertisement Corpus)								
2-5	411	250	67	26.80	16.30	0.20	0.035	471.43
3-5	189	145	41	28.28	21.69	0.25	0.073	242.47
3-10	237	172	48	27.91	20.03	0.23	0.075	206.67
5-20	107	73	27	36.99	25.23	0.30	0.141	112.77
10-20	31	22	6	27.27	19.35	0.23	0.229	4.37
10-30	40	28	8	32.14	22.50	0.26	0.247	5.26
30-40	3	2	1	50.00	33.33	0.40	0.222	80.18
30-50	3	2	1	50.00	33.33	0.40	0.222	80.18
50-100	4	3	1	33.33	25.00	0.29	0.178	60.60
100-200	**2**	**2**	**0**	**0**	**0**	-	**1.000**	-

Table 3: Comparison of experimental results with Segment Constraint on the three Engish-Hindi parallel corpora

Corpora	English corpus		Hindi corpus	
	Total words	Distinct words	Total words	Distinct words
Storybook corpus	6609	895	7606	1100
Advertisement corpus	3795	1213	4057	1198
Sentence corpus	8540	1012	9159	1152

Table 4: Experimental root word parallel corpora of English -Hindi

Different Corpora	Actual alignment in text	Alignment given by system	Correct alignment given by system	R%	P%
Advertisement corpus	323	358	253	78.32	70.68
Storybook corpus	609	546	476	78.16	87.18
Sentence corpus	4548	4548	4458	98.02	98.02

Table 5: Results of Church and Gale Algorithm for Sentence level Alignment

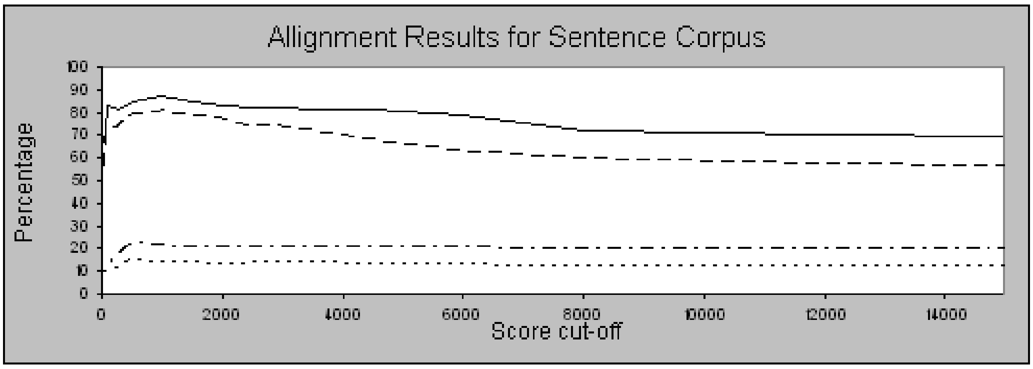

Figure 2: Alignment Results for Sentence Corpus

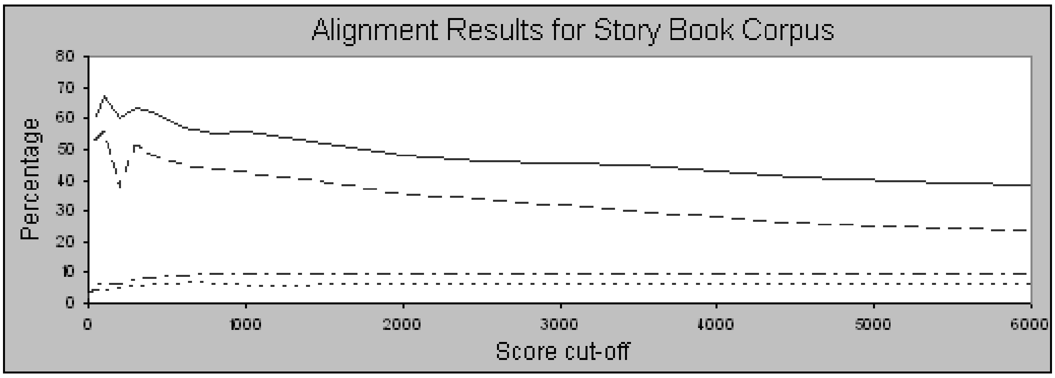

Figure 3: Alignment Results for Story Book Corpus