EFFECTIVE ESTIMATION OF CONTEXT SIMILARITY: A PROPOSED MATCHING MODEL BASED ON WEIGHTED SEMANTIC LOAD

Mehdi Mohammadi [1] and S.M. Fakhrahmad [2]

[1]Department of Computer Engineering, Sheikhbahaee University, Isfahan, Iran
mehdi.mka@gmail.com
[2]Department of Computer Engineering, Islamic Azad University, Shiraz Branch, Shiraz, Iran
mfakhrahmad@cse.shirazu.ac.ir

ABSTRACT

In this paper, we propose a new model to calculate the similarity of two sentences. The proposed scheme is based on the amount of semantic load which is shared between two sentences. Since verb is the essential part of a sentence, the main focus of the proposed model is on the verbs of two sentences. We supposed the verb as the anchor of the sentence which carries the most semantic of the sentence. The proposed model depends on part of speech (POS), the partial order of words in the sentence and the words' senses. The results by Precision and Recall are promising and benchmarks show that the proposed method improves the quality of the retrieved matched sentences.

KEYWORDS

Semantic Load, Context Similarity, Machine Translation, Example Based Machine Translation

1. INTRODUCTION

In example based machine translation, matching is a process in which similar sentences to a specific input are retrieved from a set of prepared examples. This process undoubtedly needs a similarity measurement criterion [1]. One potential solution to define the criteria would be based on the weight of each word in a sentence to complete the meaning of the sentence. If two sentences have an identical verb, their concept would be the same in a high probability. For example in the following pair of sentences A and B, the main concept of the sentence (which is "arrival") can be inferred:

A: I arrived on time.
B: He arrived late to his work.

But other words of two sentences cause to vary the meaning of two sentences. Both of the sentences have complete and independent meaning, but the concepts of the sentences are not the same. If we assume the value 1.0 as the maximum value of a sentence meaning that we can deduce, both of the above sentences can receive the score 1, but they are in two different directions. It looks like vectors as some researchers have represented the documents as vectors [2][3]. Two vectors can have the same length but be in different directions. If two vectors with

DOI : 10.5121/ijaia.2012.3301

equal size have the same direction, we say they are exactly the same. This comparison is true for sentences, too. If two sentences have the same content (such as “I arrived on time.”), they are exactly the same. But if we have two sentences (such as “I arrived on time.” and “I arrived according the schedule”), they are not exactly the same but they are equivalent. They are in the same direction and bring one concept to mind. Obviously, if we remove the verbs of two sentences, the meaning of the sentences would be unclear, whereas by removing other words, the main meaning of the sentences may change less or even will not be destroyed. So we could define a similarity measure based on whatever are equal the two sentences semantically. Two sentences are similar if the semantic load of one comes close to the semantic load of another [4]. We didn't find any clear consensus for definition of semantic load. But we achieve the similarity by defining the **shared semantic load** between two sentences. By calculating such a criterion, we could match two sentences and identify their similarity. The calculation is based on shared parts of two sentences. Complete similarity can be mapped to one and certain dissimilarity to zero. Based on this measurement, the closer to one the shared semantic load, the more similar the pair of sentences.

Content words are the main carriers of a sentence meaning rather than functional words [4]. Among the content words, verbs are ones that have a higher weight in conveying the prime object of a sentence. A verb is often defined as a word which shows an action or a state of being. The verb is the heart of a sentence - every sentence must have a verb [5]. All other structures of sentences depend on the verb. Recognizing the verb is often the most important step in understanding the meaning of a sentence. We could consider the verb of the sentence as an anchor for calculating the similarity metric. A high part of the similarity value of two sentences is obtained from the similarity of the verbs of the sentences. For simplicity, we considered the sentences which have only one main verb. The values we assign to the verbs are more than other words in the two sentences. If the stem of verbs are different, we discard comparing other words of the sentences. The similarity measurement is completed by matching the words before the matched verbs and the words after them. Thus, the two partitions besides the verb have an amount of similarity measurement inside them. Based on this, all words in the corresponding partitions in two sentences are matched, and then the whole similarity measurement is calculated based on the similarity values obtained from the matched words in two partitions. Two sentences which have a similarity value of one are exactly equal sentences. For example, in sentences A and B, some level of semantic similarity is inferable because of the same verb stem existing in the two sentences. However, the difference between the subjects and adverbs lessens the semantic similarity of the sentences.

We focus on English sentences in which word order is almost according to S-V-O[1]. Each of the two sentences is divided into three partitions such that the verb is in the middle of the other two partitions. In each partition, POS of every word is identified and tagged. Then the corresponding partitions are matched and compared. Thus, the similarity value of two sentences is composed of the matching scores obtained from each of their three partitions. The matching is performed in different levels, from exact matching to POS matching. Every matching level has its own score.
The rest of the paper is organized as follows. In section 2, we overview some related works in the literature. The proposed model is investigated in details in Section 3. In this section, the method of approximating the share of semantic load for the verb is described, too. Section 4 is devoted to the evaluation of the proposed model. Finally, Section 5 concludes the paper.

[1] Subject-Verb-Object

2. RELATED WORKS

There have been proposed different types of matching algorithms. Some popular matching approaches are Character based Matching, Word based Matching, Structure based Matching, Annotated Word-based Matching, Carroll's "Angle of similarity", Dynamic Programming Matching and so on [1] [6].

One matching algorithm is the edit distance between two strings [7]. In this algorithm, some operations such as insertion, deletion and substitution are performed to obtain a given string from a sample string. Some methods have used N-gram- based segmentation and searched them through the examples database [8].

Kfir Bar et. al. [9] have developed a matching algorithm that uses different levels of matching. They proposed six matching level, each level has a value between 0 and 1. Sumita and Iida [10] also proposed a semantic distance metric that is determined by the Most Specific Common Abstraction. The distance value is acquired from a thesaurus abstraction hierarchy. There are some attempts that are focused on semantic matching [11]. The vector representation of text units and dot product of the vectors along with considering the weight of each word results to the calculation of similarity score. In [12] also has been proposed a semantic matching procedure in which at first verb part of the input sentence and the examples are matched. Then partitioning is done in the next levels till an appropriate sub-partition is found. For this sub-partition, the exact matching is applied. For all matched examples, the distance to the input sentence is measured using a distance formula. The distance is calculated on the basis of weighted average of difference in attribute, status, gender, number, person, additional semantic, and verb category between example sentences and an input sentence.

The idea of using semantic load comes back to the work of Papageorgiou et. al. [4] [13] that has proposed a sentence level alignment algorithm by presenting a definition for semantic load of a sentence and its calculation. Their definition of Semantic Load is as the patterns of all POS tags that can be assigned to the content words of a sentence. Based on meaning preservation principle, they tried to retrieve translation examples that their semantic load can approximate the semantic load of the input sentence.

3. PROPOSED APPROACH

The composition of all consisting words of a sentence carries the meaning and the concept of the sentence. Thus, a sentence which has a complete and independent meaning would carry a complete semantic load. The semantic load of a complete sentence is shared among its parts. Unlike functional words, content words have the main role to carry the meaning of a sentence. But how much is the share of every word for the semantic load of the whole sentence? It is obvious that we cannot consider an equal value for every word's share of semantic load. Some parts of a sentence carry higher portion of the sentence meaning of compare to other parts. For instance, it is not acceptable that for a sentence such as "I read the book", every word has the same value 0.25. Obviously, the determiner here carries the lowest portion of the sentence semantic load. If the determiner is removed from the sentence, the main concept will be still understandable. If we omit the word "read" from the sentence, the result sentence "I the book" would be meaningless such that we can't understand anything from it. If the word "book" is omitted from the sentence, the resulting sentence "I read the" is incomplete and ambiguous, but the action of reading is inferable from it. It should be pointed here that compound and phrasal verbs should be treated atomically, i.e., the preposition should not be separated from the main part of the verb. For example, the preposition "off" cannot be separated from the verb "take off".

Based on this belief that the verb of a sentence carries the most portion of the semantic load (as will be proved later), the proposed method is based on dividing the sentence into three partitions: *before the verb*, *verb* and *after the verb*. This partitioning is a more general form of SVO word order. Each partition carries a share of the semantic load of a sentence. We define the *shared semantic load* of two sentences as sum of similarity level between three corresponding parts in two sentences. In order to calculate the similarity of two sentences, each of the three parts in two sentences are matched and compared correspondingly; then the similarity between each of two corresponding parts is calculated based on different matching levels. Then, the three measured similarity values are integrated in order to compute the whole similarity of the sentences under investigation.

Suppose S1 and S2 are two sentences for which the similarity value is going to be measured. First of all, each sentence is partitioned into three parts, as follows:

S1: {m words before Verb_{s1}}{ Verb_{s1}}{n words after Verb_{s1}} m, n >=0
S2: {p words before Verb_{s2}}{ Verb_{s2}}{q words after Verb_{s2}} p, q >=0
$\text{Stem}(\text{Verb}_{s1}) = \text{Stem}(\text{Verb}_{s2})$

The most important part of the matching process is to identify the verbs. Before partitioning a sentence, we identify the POS tags of every word in the sentence. The POS-tagger component identifies the correct POS of the verbs that may have Noun POS using some dependencies like the prepositions and determiners occur near that verb-form word. Suppose the input sentence is:

(a) My boss sent me a copy of this file.

Then the tagged sentence would be as below, which the word *copy* is identified as noun rather than verb:

My<N> boss<N> sent<V> me<N> a<DET> copy<N> of<Prep> this<N> file<N>.

and three partitions of this sentence would be:

My boss	sent	me a copy of this file

Similarity measure is a number between 0 and 1. The value 1 stands for the exact similarity, while 0 shows the complete difference of the sentences. Whatever the similarity measure is close to 1, two sentences would be more similar. How the similarity measure would be shared between different parts of sentence is a question that we try to answer. There is not any exact or approximate measure of semantic load shared among the parts of a sentence. As a simple heuristic, we consider an equal share of semantic load for before-verb and after-verb parts. Thus, we just have to determine the share of semantic load for the verb part. As shown in Figure 1, if we suppose x as the share of the semantic load for the verb part, then the share of semantic load for both before and after-verb parts will be ***(1-x)/2***. Figure 1 shows the partitioning a sentence and the similarity score associated to each part.

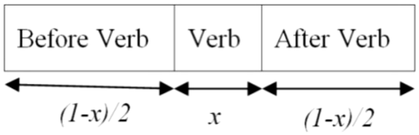


Figure 1. sentence partitioning and semantic load sharing among partitions

3.1. Approximating the share of semantic load for the verb part

To determine the share of semantic load for the verb part we arranged an experiment. In this experiment, we tested the proposed matching model using different values of the semantic load share assigned to the verb part (varying from 0.3 to 0.8 with the step of 0.1). In this experiment, we used an example set of English sentences which are the first 100,000 sentences of Europarl [14]. We also prepared an input set containing 500 sentences selected randomly from different sources. The six runs of the algorithm (using different values of the semantic load share of the verb) are applied on the input set and their matched sentences are obtained. The matched sentences are evaluated manually. The below criteria are used for evaluation:

A) The input sentence is extractable from the matched sentence by some minor changes.
B) The input sentence and the matched sentence have a shared verb stem and more changes are required to obtain input sentence from the matched sentence.
C) The input and matched sentences don’t have any shared verb, but they have some other shared words.

With this type of scoring, the effect of every verb score in six runs are calculated by counting the best matches (matches with higher scores).

In figure 2 it is demonstrated that by the verb score 0.6, the matched sentences with grade A are increased and in contrast the number of matched sentences with grade C are reduced. This guides us to choose 0.6 as the share of verb score for semantic load in similarity measurement.

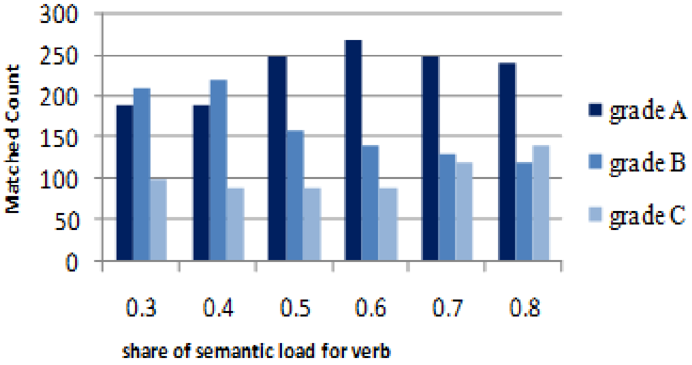


Figure 2. Comparing the best matches for different values for the share of verb score

3.2. Implementing the Matching Algorithm

In order to find the best match for an input sentence, we compare it to some sentences of the example set which have the same verb stem as the input sentence. For this purpose, the POS tags of the input sentence’s words are identified and the verb of the sentence is determined. Then, a set of sentences whose verb stem is similar to the verb stem of input sentence are retrieved from the example set. We called these retrieved examples as match candidates. The different POS tags used in this system are Noun, Verb, Auxiliary Verb, Determiner and Preposition. Then the similarity score is calculated for the input sentence and each of the match candidates. The matched candidate with the highest similarity measure is opted as the matched sentence.
When the share of the semantic load for the verb is determined as 0.6, the share of before-verb part and after-verb part will each be 0.2. This value should be distributed among the words

contained in each part of the example sentence. We refer to the distributed values as potential score (p-score) because a word may not gain this similarity score; but it is an upper bound for the similarity value of that word. Finally, each word will be assigned a contribution score denoted by c-score in this paper. The c-score is calculated for each word based on its p-score and matching level which is described in Table 1.

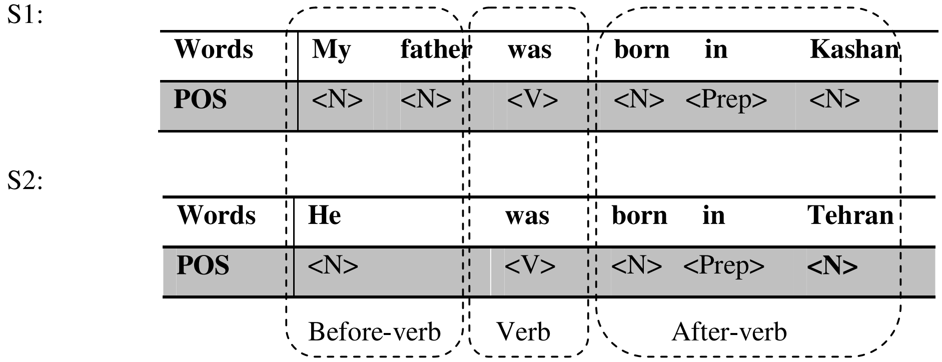
We also considered the matching level for POS tags. In this case, if two corresponding words are not exactly matched, but their POS tags are the same, a quarter of a p-score is added up to the similarity value of the sentences. On the other hand, if there is no matching between two words in POS level, no value will be added up to the similarity score of sentences. For example, if the before-verb part of a sentence consists of two words, the p-score for each would be 0.1. To calculate the c-score of each word, if each corresponding words in before-verb parts of two sentences are the same, the c-score also would be 0.1; but if they are matched in POS level, a quarter of p-score i.e. 0.025 would be assigned to the c-score of that word. As shown in Table 1, the lower the level of matching, the lower the value of c-score.

In both before-verb part and after-verb part, the corresponding words are matched and their score is added up to the similarity score of the sentences. In order to calculate the similarity value for a typical pair of sentences, firstly, the total score which is denoted by *totalScore* is initialized to zero. Then, in every step the corresponding parts are matched in turn, and the similarity measure for each part is added up to the total score. As mentioned before, we define three matching levels for each part of the sentences which are described in Table 1.

Table 1. Different levels of matching

Matching Level	**Description**
Stem	Matching in verb stem, adds up 0.6 of a complete similarity score to the similarity value of the sentences.
Exact Word	Exact matching adds up the c-score of that word to the similarity measure of the sentences.
POS	POS Matching adds up a quarter of c-score for that word to the similarity value of the sentences.

Example 1
The investigated approach is applied to the below sentences S1 and S2 as input sentence and example sentence respectively, along with the POS tags. Figure 3 demonstrates this example.



S1:

Words	My	father	was	born	in	Kashan
POS	<N>	<N>	<V>	<N>	<Prep>	<N>

S2:

Words	He	was	born	in	Tehran
POS	<N>	<V>	<N>	<Prep>	<N>

Figure 3. Partitioning and matching two sentences

Based on the shared verb stem, the share of verb in similarity measurement which is 0.6 is added to scores. In both before-verb and after-verb partitions, each word is matched to its corresponding counterpart in sentence S1 and c-score of each word is calculated. The c-score values are shown in Table 2. In this example, “He” is matched with “father” in S1 in POS level. Once we reached the beginning of the example sentence in before-verb part, the matching steps in that partition would be stopped automatically. At the after-verb partition, “born” and “in” are matched as exact matches in two sentences, but the words “Kashan” and “Tehran” are matched at POS level. Again, reaching the end of the example sentence in after-verb part, the matching steps in that partition should be stopped.

Table 2. Calculation of c-scores for an example sentence

Partions	Before-Verb	Verb	After-Verb		
Words	He	was	born	in	Tehran
c-scores	0.05	0.6	0.066	0.066	0.016

The total score of similarity comes from sum of the c-scores:

$$totalScore = 0.05 + 0.6 + 0.066 + 0.066 + 0.016 = 0.798$$

As a matter of fact, the input sentence and the matched one should have a proportional length. So the length of the sentences should be considered in calculating the similarity score. For this purpose, the similarity score of two sentences is normalized with the length ratio of the two sentences. By this normalization, two matched sentences with close lengths gain a higher score compared to the sentences with different lengths. On the other hand, the length ratio should be applied to the similarity score in such a way that when an input sentence is a sub sentence of an example sentence, their similarity score does not decrease too much. In order to avoid this side effect, the ratio of the logarithm values of both sentences’ length is used. This ratio is multiplied with the similarity score (described in the last part) to obtain the final similarity score. If the lengths of the sentences are equal, the ratio of their logarithm values would be 1, and so, their final similarity score would be just equal to the similarity score obtained in the previous step. It should be pointed that the algorithm does not allow an empty sentence to be used as the input sentence. Empty sentences cannot exist in the set of example sentences, too. The minimum length of a sentence (i.e., the number of words) should be 1. Consequently, we can formulate the final score as follows:

$$finalScore = tatalScore \times \frac{\log(\min\{length(s1), length(s2)\})}{\log(\max(length(s1), length(s2)\})} \quad (1)$$

As an example, suppose that the length of input sentence is 32. Also assume that there is a sentence in the example set which contains the input sentence and its length is 128. In this case, the value of $\frac{\log(32)}{\log(128)}(= 0.71)$ would be multiplied to similarity score that is better than direct ratio of two lengths i.e. 0.25. In this algorithm, logarithm could be calculated in any base.

To complete the example 1, its *finalScore* could be calculated based on formula (1). The length of S1 is 6 and the length of S2 is 5 words:

$$finalScore_{example1} = 0.798 \times \frac{\log(5)}{\log(6)} = 0.798 \times 0.898 = 0.716$$

A partial order is supported by the proposed method. When the same verb roots in two sentences are matched, it is the beginning of an approximate order of other corresponding parts in two sentences. In a normal case for a same verb stem in two sentences, the before-verb part is a place which the subject can occur, whereas the after-verb part is a place that object and other words can appear.

4. Evaluation and Experimental Results

There are several criteria for evaluation of the information retrieval algorithms. Among them, Recall and Precision are frequently standard ones [15]. Precision is defined as the number of correct matches divided by the total number of matches (correct or incorrect) and Recall is the number of correct matches divided by the number of all sentences in the example set that are similar to the input sentence. Detecting the correct matches is performed manually.

In order to evaluate the matching method, we selected a set of examples containing about 500 sentences from English Wikipedia, randomly. We also constructed a set of input sentences consisting of 200 sentences and also a reference set of correct matches consisting of 200 sentences. This reference set is prepared manually in a way that contains the most similar example sentence to each input sentence. Therefore, the reference set is a subset of the example set and contains 200 sentences of the example sentences. The proposed method is applied to the input sentences and a matched sentence for each input sentence is retrieved. In order to compare the proposed metric to some standard similarity metrics, the Cosine and Jaccard methods are applied to the input set in turn, and the matched sentences are obtained in each case. Then, the precision is calculated for three methods. The results are shown in Table 3.

Table 3. Comparing the precision of proposed method

Method	**Correct Matches**	**Total Matches**	**Precision**	**Recall**
Jaccard	83	200	41%	41%
Cosine	82	200	41%	41%
Proposed	188	200	94%	94%

The results show that the proposed matching method has a higher precision compared to the standard methods used in Information retrieval methods. One drawback of Jaccard and Cosine approaches is that they view a sentence as a bag-of-words, which consider only the common words in the two sentences and do not care about words' order. Considering words' order is one of the most important points of the proposed method having a high effect on its efficiency. For further illustration, consider three example sentences existing in the example set, as shown in Table 4. Assuming that the input sentence is "Can you open the door?", Table 5 shows the retrieved match for this sentence using different similarity metrics.

Table 4. Some sentences used in example set

Example sentence
please open the door
the door is open for you
the open door is wooden

Table 5. Standard approaches versus proposed method to retrieve a match

Method	Retrieved Match
Jaccard	the door is open for you
Cosine	the door is open for you
Proposed	please open the door

As it is clear from Table 5, Jaccard and Cosine similarity metrics retrieve the same sentence as the matched sentence, while it is obvious that the best match has been obtained by the proposed method.

It shows that adding grammatical information along with the words order to the matching approach has a direct effect on the accuracy and precision of the matching method. In other words, in order to retrieve the best match, grammatical similarity should be exploited in addition to the context similarity.

5. CONCLUSION AND FUTURE WORKS

In this paper, a model to match input sentences against example sentences is proposed and exploited in a matching component. In this model, based on closeness of the semantic load of two comparing sentences, the most similar sentence to the input sentence is retrieved from the example set. The stress on the verb along with the special storage structure of examples which has defined an index on verbs of example sentences demonstrated a more precise behavior of proposed model rather than other standard retrieval methods. The approximate words order is considered in the proposed model. In other words, the verb of sentence is seen as an anchor that there are some words before it and there are some other words after it. The matching process goes forward based on this approximate order. This causes that fidelity to the original text would be increased in translation by selecting the most similar example sentence to input sentence.

The evaluation results showed that prioritizing the verb for matching increases the accuracy of the matching method so that its precision is twice of standard methods like Jaccard and Cosine. The current research is done for simple sentences. We intend to cover compound and multipart sentences in future works. The POS tags also should be in more variation. More research also is needed for determining the share of semantic load for the before-verb and the after-verb parts which we considered them equal in this paper. Synonym verbs with two different stems are also the issue which we intend to take into account of the similarity measurement.

REFERENCES

[1] Somers, H., (1999) "Review Article: Example-based machine translation", Machine Translation, Vol. 14, pp113-157.

[2] Turney, P. D. & Pantel, P., (2010) "From Frequency to Meaning: Vector Space Models of Semantics", Artificial Inteligence Research, Vol. 37, pp 141-188.

[3] Cranisa, L. & Papageorgiou, H. & Piperidis, S., (1994) "A Matching technique in Example Based Machine Translation". Proceedings of the 15th conference on Computational linguistics,. Kyoto, Japan, pp. 100-104

[4] Papageorgiou, H. & Cranias, L. & Piperidis, S., (1994) "Automatic Alignment in Parallel Corpora", ACL '94 Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pp 334-336.

[5] Lauren K. & Duncan, M.D, (2007) When Words Collide (7th ed.),Wadsworth Publishing.

[6] Sumita, E., (2003) "Example-based machine translation using DP-matching between word sequences". In Recent Advances in Example-Based Machine Translation. Dordrecht, Kluwer Academic Publishers. pp 189-209.

[7] Vertan, C. & Martin, V.E., (2005) "Experiments with matching algorithms in example based machine translation". Modern Approches in Translation Technologies, pp 42-45.

[8] Way, A. & Gough, N., (2003) "wEBMT: Developing and Validating an Example Based MachineTranslation System Using the World Wide Web". Computational Linguistics , Vol. 29 No. 3, pp 421-458.

[9] Bar, K. & Choueka, Y. & Dershowitz N., (2008) "Matching Phrases for Arabic-to-English Example-Based Translation", Language, Culture, Computation: Studies in Honor of Yaacov Choueka. Springer-Verlag.

[10] Sumita, E. & Iida H., (1991) "Experiments and Prospects of Example-based Machine Translation". Proceedings of the 29th Annual Meeting of the ACL, pp 185-192.

[11] Gupta, D. & Chatterjee, N., (2002) "Study of similarity and its measurement for English to Hindi EBMT", STRANDS'02. Kanpur.

[12] Jain, R., (1995) "HEBMT: A Hybrid Example-Based Approach for Machine" PhD Thesis, I.I.T, Kanpur.

[13] Piperidis, S. & Papageorgiou, H. & Demiros, I. & Malavazos, C. & Triantafyllou, Y., (1998) "A Framework for Example-Based Translation Aid Tools", Proceedings of the Panhellenic Conference on New Information Technology-(NIT'98), pp 269-278.

[14] Koehn, P., (2005) "Europarl: a parallel corpus for statistical machine translation" MT summit X, the tenth machine translation summit, pp 79-86.

[15] Ahrengerg, L. & Merkel, M. & Hein, A.S., & Tiedmann, J., (2000) "Evaluation of word alignment systems" LREC2000, pp 1255-1261.