Semantic Hierarchical Document Signature For Determining Sentence Similarity

Sukanya Manna and Tom Gedeon

Abstract— **In this paper, we present a new approach that incorporates semantic information from a document, in the form of *Hierarchical Document Signature* (HDS), to measure semantic similarity between sentences. Due to variability of expressions of natural language, it is very essential to exploit the semantic properties of a document to accurately identify semantically similar sentences since sentences conveying the same fact or concept may be composed lexically and syntactically different. Inversely, sentences which are lexically common may not necessarily convey the same meaning. This poses a significant impact on many text mining applications performance where sentence-level judgment is involved. Our HDS uses the natural hierarchy of the document and represents it in a modularized form of document level to sentence level, sentence to word level; aggregating similarity components at the lower levels and propagating them to the next higher level to produce the final similarity between sentences. The evaluation of our HDS model has shown that it resembles the decision making process as done by human to a greater extent than different vector space models which only uses 'bag of words' concept.**

I. Introduction

In this paper, we propose an application of hierarchical document signature (HDS) [1], extension of fuzzy signature [2], [3], [4] that takes into account semantic structure of sentences to measure sentences similarity. Traditionally, sentences are transformed into a 'bag of words' for sentence similarity computation as in cosine similarity, Jaccard's coefficient and so on. This results in "semantic loss" because semantic contextual senses of the sentences are discarded. In particular, this has a crucial consequence to the identification of semantic equivalence. Our proposed method aims to deal with this issue by utilizing semantic similarity of constituent words in the sentences and then using that information to find the overall similarity between pairs of sentences using HDS structure.

It is known from grammatical perspective that words in a sentence can belong to four main parts of speech; noun, verb, adjective, and adverb. Each impart different information to the context of a sentence. Two different words can impart similar meaning to a context; likewise a word can also express different meanings in different contexts. For example, the word 'hit' and 'crash' can be used synonymously when it comes to the context of 'collision with something', when both the words are verb. On the other hand, the noun representation of 'hit' can mean 'success' as in case of 'songs or movies becoming a hit'. In this work, we try to catch this feature of natural language in finding similarity between sentences.

Using HDS we modularize a document in a hierarchical manner; document level to sentence level, sentence level to parts of speech level; then parts of speech to word level. The basic computation of semantic similarity begins at word level, from where the similarity score is propagated to the next higher level using proper aggregation. This HDS is designed in such a way, that any kind of text analysis tasks which needs step-wise decision making process can be also be interpreted; provided the levels and aggregation needs to be tuned based on those specific applications.

The main challenge of this work is to formulate a sentence similarity measure which uses fuzzy logic for decision making in finding similar sentences in the similar way as human judgements. This method is mainly formulated to analyze single documents or small sets of documents which requires higher precision of results for its application such as legal report analysis, investigation/witness related documents, technical reports; unlike traditional information retrieval indexing or search purpose.

The following sections of this paper explain in details about the semantic hierarchical document signature and how it is applied to find semantic sentence similarity along with some evaluations.

II. Related work on sentence similarity

The issue of measuring similarity of sentences is gaining more attention from various research communities. Necessity of text similarity may vary depending on the application domains, many of them share a common goal of matching up semantically similar sentences. Various techniques have been proposed to perform sentence similarity. First, probabilistic approaches have been adopted to identify topically related sentences [5], [6], [7] in sentence retrieval application. Next, several unsupervised approaches have been proposed for paraphrase recognition tasks [8], [9], [10], [11]. Recently, natural language processing community has increasingly focused on developing NLP systems to recognize entailment between sentences [12]. For this task, systems that employ extensive linguistic tools, such as logical inference engine and anaphora resolution [13], [14], have started to show significant improvement in result over relatively shallower approaches. Nevertheless, this comes with a trade off in computational cost which makes comprehensive NLP systems currently impractical for a large text collection. We are motivated by [15], [16], and [17]. [17] incorporates semantic

The authors are with the *Information and Human Centered Computing group*, School of Computer Science, The Australian National University, ACT 0200, Australia; email:{sukanya.manna,tom.gedeon}@anu.edu.au

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

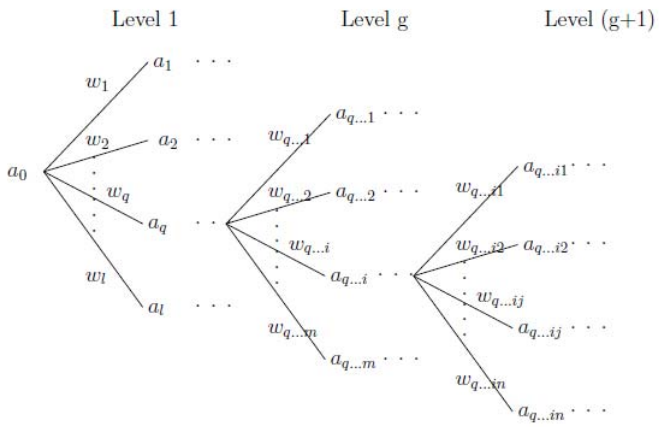

Fig. 1. Illustration of an arbitrary signature

structure of sentences, in a form of verb-argument structure, to measure semantic similarity between sentences.

Our work aims to address some of the shortcoming of existing text similarity measures but instead of considering semantic roles of sentences, we find out semantic similarity at word levels based on their parts of speech, and then aggregating and propagating the similarity values from word level to higher level of HDS to get final sentence similarity score.

III. Hierarchical Fuzzy Signature

Fuzzy signatures [2], [3] are extension of vector valued fuzzy sets [18], [3] which can describe, compare and classify objects with complex structure and interdependent features. The hierarchical organization of fuzzy signatures express the structural complexity of a problem. The local preference relations among the hierarchies and sub-branches of a fuzzy signature can be used to approximate the global preference relation of a decision problem.

Definition 1: Fuzzy Signature is a VVFS, where each vector component is another VVFS (branch) or a atomic value (leaf), and denoted by,

$$A : X \rightarrow [a_i]_{i=1}^{k} \left(\equiv \prod_{i=1}^{k} a_i \right). \tag{1}$$

$$where\ a_i = \begin{cases} [a_{ij}]_{j=1}^{k_i} & ;if\ branch \\ [0,1] & ;if\ leaf \end{cases}$$

and Π describes the Cartesian product.

IV. Hierarchical Document Signature

The hierarchical document signature (HDS) [1] is a special type of fuzzy signature (FS) [2], [3] which is used for document analysis purpose. In this case, the natural hierarchy of a document is maintained through the structure of HDS; document level to sentence level, sentence level to word level. HDS, like fuzzy signatures, can describe, compare and classify objects with complex structure and interdependent features. The hierarchical organization of HDS express the structural complexity of a problem. The local preference relations among the hierarchies and sub-branches of a document signature can be used to approximate the global preference relation of a decision problem.

Using HDS we can model sparse and hierarchically correlated data with the help of hierarchically structured vectorial fuzzy sets [19] (which are constructed using the information extracted from the document) and a set of not-necessarily homogenous and hierarchically organized aggregation functions. The set of aggregation functions [3] map the different universes of discourse of the hierarchical fuzzy signature structure, from lower branches to the higher branches. We argue that these properties help fuzzy signatures to model problems similar to the nature of human comprehensible hierarchical approaches to problem solving; and so in case of HDS.

An important advantage of the fuzzy signature concept is that it can be used to compare degree of similarity or dissimilarity of two slightly different objects, which have the same fuzzy signature skeleton. We exploited this feature of FS to find the sentence level semantic similarity of a document by propagating the information from the word level to document level. Thus we see that HDS is a modified form of fuzzy signatures in [3].

Definition 2: Hierarchical Document Signature (HDS) can be defined as a special class of Fuzzy signature which contains the natural hierarchy of a document in the form of vectors. Broadly it consists of three vectors which are document vector, sentence vector, and word vector; D_v, S_v, and W_v; which is the inbuilt hierarchy of the document itself.

$$d_i = [s_j] \in S_v;\ d_i \in D_v \tag{2}$$

$$s_j = [w_l] \in W_v;\ s_j \in S_v \tag{3}$$

Lemma 1: HDS can also contain further sub-branches at any level based on different applications of document analysis for storing specific information. Sentence vector can be further branched into attribute vector, where words present in the sentence are grouped based on some user defined attributes, A_v. Word vector can also be further extended into feature vector, F_v, which is mainly used for storing different features of the document (e.g. word frequency, word location in the sentence etc) at word level for the whole analysis.

$$d_i = [s_j] \in S_v;\ d_i \in D_v \tag{4}$$

$$s_j = [a_k] \in A_v;\ s_j \in S_v \tag{5}$$

$$a_k = [w_l] \in W_v;\ a_k \in A_v \tag{6}$$

$$w_l = [f_m] \in F_v;\ f_m \in F_v \tag{7}$$

In this case, for exploiting semantic information of a document, the attribute vector is 'Parts of Speech (POS)' unlike in [1]; having four basic POS namely, *noun*, *verb*, *adjective*, and *adverb*. Here, we have further subclassified noun into *proper noun* and *common noun*. Proper noun tag contains all the information regarding specific names of place, person, organization, time and so on and the common noun contains all the generic nouns present in a particular sentence.

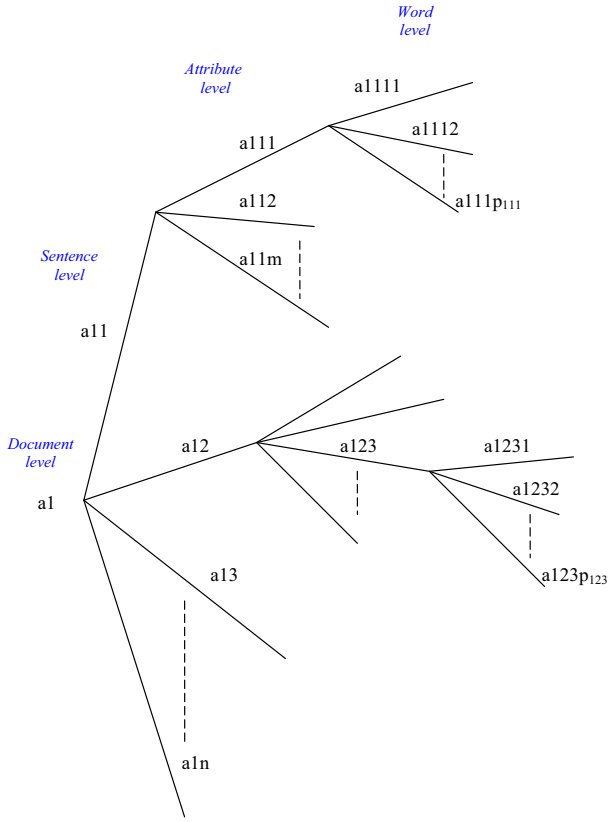

Fig. 2. Generic illustration of Hierarchical Document Signature

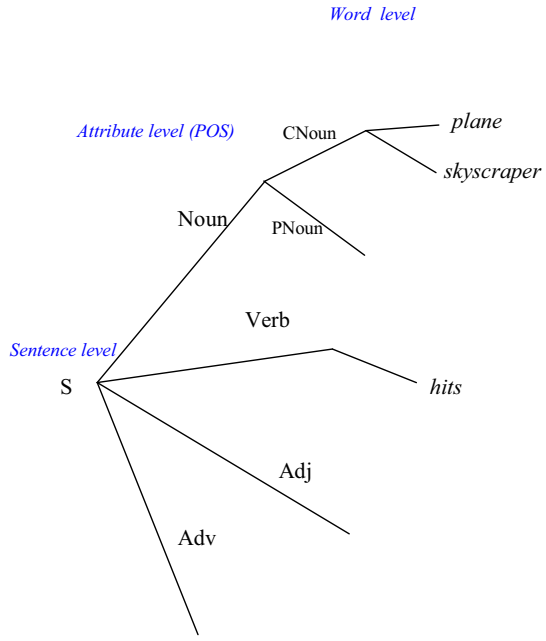

Fig. 3. Representation of a sentence (A plane hits a skyscraper) in the Hierarchical Document Signature

We discuss here more problem specific HDS. The levels and branches are flexible according to different applications. In fig.2, $a1$ represents a document at *document level*. A document is now segmented into n sentences, which we denote by $a11$ to $a1n$. This is the *sentence level* of the signature we developed. Next is the *attribute level*, which basically classifies the words of the sentences into their corresponding major *parts of speech*, namely *noun*, *verb*, *adjective* and *adverb*. So, for each sentence, we have m different attributes, which is fixed for each document signature. In general m is a integer, but it is constant with specific application for the ease of comparison. as shown in fig. 3. Each of these attributes per sentence is presented as $a111$ to $a11m$ for sentence 1, similarly, $a121$ to $a12m$ for the second sentence and so on. Now each attribute of each sentence has words, and this level is called *word level*. There can be any number of words in each attributes (POS). Here, p_{111} is the maximum number of words for attribute $a111$. Suppose we have two signatures for sentence 1 and sentence 2 of a document. Now, for signature 1, if $m = 2$ (let the attributes be proper noun and verb), and for signature 2, $m = 4$ (let the attributes be proper noun, verb, adjective, adverb), then for two signatures it is tough to compare. It make it computationally very expensive, at the same time it loses practicality. As in this example, we can clearly see that there is no point of comparing POS proper noun with POS verb of the two signatures. It is meaningless to compare different parts of speech, as a single word can impart different sense to a context in different parts of speech. Thus it is necessary to compare the same POS of different sentences for finding similarity between them.

V. Computations at different levels HDS to find sentence similarity

In general, two sentences are said to be semantically equivalent if they share similar meaning or sense based on their context. Humans can easily identify sentences which are similar; but when we try to mimic this automatically, it becomes much complicated and difficult because of the basic construct of natural language; where same words impart different meaning in different context, and different words can also impart similar meaning to a particular context. For this, a lexical dictionary like WordNet [20] is required which provides semantic information about the words and their senses in different parts of speech and defining different relations it has with other words of the same family.

Thus in this paper, we use HDS to modularize a document into sentences and then into words by classifying words into their corresponding parts of speech. Then by finding semantic similarity of words at the word level of same parts of speech of a pair of sentences and then propagating the similarity score to the next higher level with suitable aggregation giving the final similarity score between two sentences.

A. Word level: semantic similarity between words

In this part, we explain how we formulated the semantic word similarity using fuzzy inference system using WordNet

as a lexical dictionary for obtaining sematic information of the words being considered.

1) Role of WordNet in semantic information: WordNet is a lexical online database for the English language [20]. It groups English words into sets of synonyms called *synsets*, provides short, general definitions (called glosses), and records the various semantic relations between these synonym sets. The main purpose is to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications.

WordNet distinguishes between nouns, verbs, adjectives and adverbs. Every synset contains a group of synonymous words or collocations; different senses of a word are represented in the form of different synsets. Each synset os provided with a short meaning called *gloss*. Most synsets are connected to other synsets via a number of semantic relations. These relations vary based on the type of word being considered, and can be hypernym, hyponym, holonym, meronym, and troponym.

WordNet also provides the polysemy count of a word i.e., the number of synsets that contain the word. If a word participates in several synsets (i.e. has several senses) then typically some senses are much more common than others. WordNet quantifies this by the frequency score: in which several sample texts have all words semantically tagged with the corresponding synset, and then a count provided indicating how often a word appears in a specific sense.

For any kind of semantic analysis job, researchers either use WordNet or Rojet's thesaurus to formulate their models.

2) Fuzzy word similarity (PGMeasure): In this section, we present a fuzzy word similarity measure. We use two different similarity measures (gloss overlap and path based measure) as input to our fuzzy inference system and as an output we have PGMeasure, where information of both the types are fused based on the rule base created for the similarity scores.

a) Using glosses of related senses: Word forms from the definitions ("glosses") in WordNet's synsets are manually linked to the context-appropriate sense in WordNet. We have seen that in [21] and [22], they have used glosses to find similarity between concepts. Here, we used [21]'s ideas but computed Jaccard's coefficient [23] of only glosses among different sentences of word pair considered to find semantic similarity between them.

Let us consider two words w_i^p and w_j^p, $i, j \in W$, and $p \in P$; where W is the set of words, and P is a set of parts of speech like *noun*, *verb*, *adjective*, and *adverb* respectively. Let word w_i^p has $|K|$ synsets and w_j^p has $|L|$ synsets which can be represented by $S_k^{w_i^p}$ and $S_l^{w_j^p}$ respectively, $k \in [1, |K|]$ and $l \in [1, |L|]$. Now we calculate *gloss overlap* as sim_{gloss} by

$$sim_{gloss}(w_{i_k}^p, w_{j_l}^p) = JaccardCoef(S_k^{w_i^p}, S_l^{w_j^p}) \quad (8)$$

b) Path based similarity scores: In this section we present a method proposed by Lin [24] which is basically corpus based approaches but also uses edge/path information for finding the similarity between two concepts present in the taxonomical hierarchy of any kind of lexical dictionary using corpus information.

TABLE I
CO-ORDINATES OF DIFFERENT MEMBERSHIP FUNCTIONS OF FUZZY SETS USED HERE

Low_trapez	(0, 0)	(0, 1)	(0.3, 1)	(0.4, 0)
Medium_trapez	(0.3, 0)	(0.4, 1)	(0.6, 1)	(0.7, 0)
High_trapez	(0.6, 0)	(0.8, 1)	(1, 1)	(0, 1)
Low_tri	(0, 0)	(0.3, 1)	(0.6, 0)	-
Medium_tri	(0.3, 0)	(0.55, 1)	(0.8, 0)	-
High_tri	(0.6, 0)	(0.8, 1)	(1, 0)	-

Lin's model: Lin [24] used a similarity model, which uses edge counting method as well as information content of the concepts to find the similarity. So, we rather prefer calling it to be *path measure* as it exploits taxonomical hierarchy of the WordNet as well as corpus based statistical measure for computation. This can be represented as,

$$related_{lin}(c_1, c_2) = \frac{2.IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} \quad (9)$$

where c_1 and c_2 are two concepts whose relatedness we are tending to find, IC determines the information content of a concept and $lcs(c_1, c_2)$ finds the lowest common subsuming concept of concepts c_1 and c_2.

Now in our case, we replace the concepts c_1, c_2 by words w_i^p and w_j^p, $i, j \in W$, and $p \in P$; where W is the set of words, and P is a set of parts of speech like *noun*, *verb*, *adjective*, and *adverb* respectively. Let word w_i^p has $|K|$ synsets and w_j^p has $|L|$ synsets which can be represented by $S_k^{w_i^p}$ and $S_l^{w_j^p}$ respectively, $k \in [1, |K|]$ and $l \in [1, |L|]$. We named this measure as *path measure* as sim_{path} by

$$sim_{path}(w_{i_k}^p, w_{j_l}^p) = related_{lin}(S_k^{w_i^p}, S_l^{w_j^p}) \quad (10)$$

c) Path-Gloss measure (PGMeasure) using FIS: We combine *Path measure*, especially the measure proposed by Lin et al., and *Gloss overlap* to create *PGMeasure* using FIS. For computation of this, we create two input fuzzy sets; one for path measure and the other for gloss overlap with three membership functions in each case based on *low similarity*, *medium similarity*, and *high similarity*, signifying to what extent two words are similar or related to each other.

Determination of input fuzzy sets We use same notations for words as used in the previous equations (8) and (10). Now, let us consider, two input fuzzy sets, $pathMeasure$ and $glossOverlap$. We used trapezoidal membership functions of *low (L)*, *medium (M)*, and *high (H)* in both the cases. The co-ordinates (x, y) for the trapezoidal membership functions are shown in table.I.

For $pathMeasure$ we define three membership functions *low*, *medium*, and *high* with corresponding membership values $\mu_L(sim_{path}(w_{i_k}^p, w_{j_l}^p))$, $\mu_M(sim_{path}(w_{i_k}^p, w_{j_l}^p))$, and

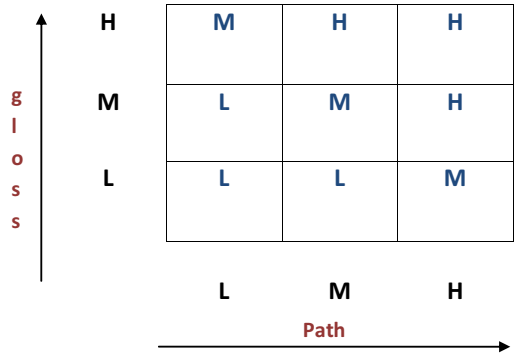

Fig. 4. Rules for FIS for obtaining PGMeasure of word similarity

$\mu_H(sim_{path}(w^p_{i_k}, w^p_{j_l})) \in [0,1]$. $sim_{path}(w^p_{i_k}, w^p_{j_l})$ is computed using (10).

Likewise, for the fuzzy set $glossOverlap$, we define three membership functions *low*, *medium*, and *high* with corresponding membership values $\mu_L(sim_{gloss}(w^p_{i_k}, w^p_{j_l}))$, $\mu_M(sim_{gloss}(w^p_{i_k}, w^p_{j_l}))$, and $\mu_H(sim_{gloss}(w^p_{i_k}, w^p_{j_l})) \in [0,1]$. $sim_{gloss}(w^p_{i_k}, w^p_{j_l})$ is computed using (8).

d) Fuzzy rule-bases: We present here the rules generated for our FIS. The rules are decided manually based on expert's understanding about the similarity measures.

e) Determining output fuzzy set: We used triangular membership functions as output fuzzy set. Triangular is used over trapezoidal in order to overcome the 'centre of gravity (COG)' calculation approximation in the latter i.e, if both the inputs are 1 then we expect the output value to be 1 as well. But due to approximation of COG in trapezoidal function, it gives the value 0.84. We can avoid this feature using triangular function in the output. Now, $\mu_L(sim_{PG}(w^p_{i_k}, w^p_{j_l}))$, $\mu_M(sim_{PG}(w^p_{i_k}, w^p_{j_l}))$, and $\mu_H(sim_{PG}(w^p_{i_k}, w^p_{j_l})) \in [0,1]$ are the corresponding membership values for the output fuzzy set PGMeasure with membership functions low, medium and high respectively. The co-ordinates of the triangular membership functions are defined in table.I. Now, $sim_{PG}(S_k^{w_i^p}, S_l^{w_j^p})$ is the defuzzified output of the the two input fuzzy sets for all combination of synsets of the word pairs w_i^p and w_j^p. Thus, the final fuzzy word similarity score of a word pair is hence computed by,

$$sim_{PGMeasure}(w_i^p, w_j^p) = Max[sim_{PG}(S_k^{w_i^p}, S_l^{w_j^p})] \quad (11)$$

which is the maximum of all the defuzzified values of all possible combinations of synsets of the pair of words considered.

B. Computation at POS level or attribute level

At the word level, we compute the semantic similarity of words belonging to same parts of speech in the pair of sentences being considered. Let us consider two sentences, s_1 and s_2. s_1 be "A plane hits a skyscraper" and s_2 be "A plane crashed into a tall building". Now, we modularize each sentence and store their respective information using HDS. For the first sentence, we have only two different POS, noun and verb. Both the nouns are common noun in this case. For the second sentence, we have noun (common noun), verb and adjective as corresponding POS of the words present. We seek for similarity of sentence 1 with respect to sentence 2. So, each of the nouns of s_1 are compared against all the nouns of s_2, and the maximum similarity score is assigned against each of the nouns with respect to the ones of s_2. Such as, we take, $SimScore_{w_i^{cnoun}} = max[sim_{PGMeasure}(w_i^{cnoun}), w_j^{cnoun}]$, where $i \in |cnoun|_{s_1}$ and $j \in |cnoun|_{s_2}$. This is how the similarity values are stored against each of the words at word level after initial comparison with the other sentence. Then *max* aggregation over the similarity scores of the words of each parts of speech is again performed to get the scores at nodes a111, a112 ... a11m as in 2 i.e., $pos^p_{s_1} = max[SimScore_{w_i^p}]$, where $p \in |pos|_{s_1}$, which is actually the number if parts of speech in sentence 1 and $i \in |p|_{s_1}$ except for noun where we take average of both proper and common noun are present otherwise we follow the same rule.

C. Computation at sentence level

Once we obtain the similarity scores for each POS we then perform two different aggregations. We find *max* of $pos^p_{s_1}$ and *mean* over the same set. Then to compute the final similarity of two sentences we take *mean* of max and mean scores already computed. Therefore we get the final similarity scores between two sentences.

VI. Experimental Evaluation

A. Data set

For this experiment, we used standard dataset of the Microsoft paraphrase corpus and an sample example to show the performance of semantic HDS. Since our HDS involves the integration of WordNet as well as Stanford parts of speech tagger[1], some noise has entered in the process as none of these online available NLP tools are fully accurate.

f) Sample example: Here we present a sample set of sentences, where the first two sentences expresses similar meaning and the other two also pair up with similar sense of the context.

1) A plane hits a skyscraper.
2) A plane crashed into a tall building.
3) People gathered to find out the cause.
4) Reporters arrived to collect information about the crash.

g) The Microsoft Paraphrase Corpus: In 2005, Microsoft researchers Dolan, Brocket, and Quirck [9] published the first paraphrase corpus containing 5801 pairs of sentences with 3900 tagged as semantically equivalent or true paraphrases. Sentences were obtained from massive parallel news sources and tagged by 3 human raters according to guidelines described in [9]. We will refer to this corpus as the label MSRPC. We have divided this data set in random order in chunks of 50 pairs of sentences for ease of representation.

[1] http://nlp.stanford.edu/software/tagger.shtml

B. Evaluation setting

We set here similar evaluation settings like [25]. In this case, we evaluate the similarities computed with respect to the benchmark similarity values provided by human in the data set used. We use the same metrices as used by [25] which are namely *recall*, *precision*, *F1*, *accuracy*, *rejection*, and *f1* respectively.

Recall is a proportion of correctly predicted similar sentences compared to all similar sentences. **Precision** is a proportion of correctly predicted similar sentences compared to all predicted similar sentences. **F1** is a uniform harmonic mean of precision and recall. **Rejection** is a proportion of correctly predicted dissimilar sentences compared to all dissimilar sentences. **Accuracy** is a proportion of all correctly predicted sentences compared to all sentences. Lastly, we define **f1** as a uniform harmonic mean of rejection and recall. A scoring threshold for positive pairs is defined at 0.5 as it is used in the literature [25].

C. Sentence similarities

We have explained here two different types of sentence similarities which have been used for evaluation. The first one is the cosine similarity which is basically vector space model and works on only 'bag of words' concept. The second one is semantic similarity computed by HDS which uses semantic information of the text along with fuzzy aggregations to compute the similarity.

h) Cosine similarity: Cosine similarity[2] is a measure of similarity between two vectors of n dimensions by finding the cosine of the angle between them, often used to compare documents in text mining. In addition, it is used to measure cohesion within clusters in the field of Data Mining [26]. Given two vectors of attributes, A and B, the cosine similarity, $similarity_{cosine}$,θ is represented using a dot product and magnitude as

$$similarity_{cosine} = cos(\theta) = \frac{A.B}{\|A\|\|B\|} \tag{12}$$

For text matching, the attribute vectors A and B are usually the term frequency vectors of the documents. The cosine similarity can be seen as a method of normalizing document length during comparison. The resulting similarity ranges from -1 meaning exactly opposite, to 1 meaning exactly the same, with 0 indicating independence, and in-between values indicating intermediate similarity or dissimilarity. In the case of information retrieval, the cosine similarity of two documents will range from 0 to 1, since the term frequencies (tf-idf weights) cannot be negative.

i) Similarity calculation using HDS: In sec.V, we explained step-wise computation of finding sematic similarities between two sentences. We modularized a document into sentences, and then further grouped the words of the sentences based on their POS. Using fuzzy word similarity measure (PGMeasure) we calculated similarity between two words and assigned them as the similarity score for each word in a sentences with respective POS. With different aggregations at different levels, we finally obtained semantic similarity between two sentences. The similarity value of 0 shows the sentences are not at all related and 1 means they are identical.

[2]http://www10.org/cdrom/papers/519/node12.html

TABLE II

SIMILARITY SCORES OF SENTENCE PAIRS USING SAMPLE EXAMPLE

SentID	SentID	Cosine Sim	HDS Sim
1	2	1.0	0.42
1	3	0.0	0.25
1	4	0.0	0.25
2	3	0.0	0.23
2	4	1.0	0.23
3	4	0.0	0.42

TABLE III

EVALUATION METRIC OF SENTENCE SIMILARITY

Methods	recall	precision	F1	accuracy	rejection	f1
Cosine	1.0	0.65	0.79	0.65	0.0	0.0
HDS	0.75	0.67	0.71	0.59	0.10	0.18

VII. RESULTS AND DISCUSSION

A. Results

j) Sample example: In the data set we have shown our sample example with 4 sentences. Here we present the similarities obtained using HDS of those sentences in table.II.

If we look at the real sample example, we can see that sentence pair 1 and 2 expresses similar meaning, and again sentence pair 3 and 4 expresses similar meaning. This is exactly what we have captured in table.II. Cosine similarity on the other hand has predicted the first one correctly, but for the second pair it could not identify the underlying semantic similarity. These are the main areas where HDS is mainly used for.

k) Using annotated MSR corpus: We used MSR corpus for evaluation of our method. We also used cosine similarity as a benchmark.

In table.III, we can see that our method has higher precision than cosine similarity. But there is minor drop in performance due to wrongly tagged words done by the parser used. This is detected when the tagging was checked manually. There is slight rejection of 0.1 in the similarity findings. In the data set, the humans assigned either 1 or 0 for similar or dissimilar sentences. But this is not very appropriate because similarity measurements when done by humans, can have certain level of uncertainty, which can be captured using fuzzy methods. This is actually done by our method and the snapshot is shown in table.IV.

B. WordNet coverage and pitfalls

The effectiveness of linguistic measures depends on a heuristic to compute semantic similarity between words as well as the comprehensiveness of the lexical resource. As WordNet is used as a primary lexical resource in this study,

TABLE IV
SNAPSHOT OF RESULTS WITH MSRPC

Sent1ID	Sent2ID	HumanScore	CosineSim	HDS Sim
3354381	3354396	0	1.0	0.37
1390995	1391183	1	0.99	0.5
2201401	2201285	0	1.0	0.55
...	...	...	...	...

its comprehensiveness is determined by the proportion of words in the text collections that are covered by its knowledge base. In general, a major criticism of WordNet-based similarity measures is in its limited word coverage to handle a large text collection, particularly on the named entities coverage. The percentage of word coverage in WordNet decreases as the size of test collection and vocabulary space increases. Thus, the effectiveness of linguistic measures is likely to be effected because word-to-word similarity calculation will inevitably produce many "misses". One solution is to resort to approaches that utilize other knowledge resources, such as Wikipedia [27] or web search results [28], to derive semantic similarity between words.

VIII. CONCLUSION

In this paper we presented a sentence similarity approach using Hierarchical Document Signature. This uses semantic information from the document using WordNet to find the underlying semantic information among words. This feature is simply discarded by vector space models like cosine similarity which uses only 'bag of words'. This works better only when it finds exact match in words on the sentences, otherwise it is not that useful. This is shown in the evaluations using sample example. Thus HDS performs better in finding semantic similarity in sentences with a higher precision. Due to limitations in WordNet, there were many words which were not a valid entry in its database; as a result of which there is effect on the overall performance as well. Besides this, the difference in membership functions of the output fuzzy set has a great impact on the final similarity results of the word pairs. Not only this, the standard POS taggers are not perfect enough to tag sentences properly. So, noise has entered in the process of tagging which degraded the overall performance when worked on MSRP corpus. As a future work, we aim to focuss on these aspects of NLP using fuzzy methods to reduce noise in using HDS, at the same time we will tune the membership functions of FIS for further refinement of the fuzzy outputs using different machine learning methods.

REFERENCES

[1] S. Manna, B. Mendis, and T. Gedeon, "Hierarchical document signature: A specialized application of fuzzy signature for document computing."

[2] B. S. U. Mendis, T. D. Gedeon, and L. T. Kczy, "Investigation of aggregation in fuzzy signatures," in *3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore*, vol. CD ROM, 2005.

[3] B. S. U. Mendis, "Fuzzy signatures: Hierarchical fuzzy systems and alpplications (phd thesis)," Ph.D. dissertation, College of Engineering and Computer Science, The Australian National University, Australia, 2008.

[4] B. S. U. Mendis and T. D. Gedeon, "Fuzzy rough signatures," *FUZZ-IEEE 2009 International Conference on Fuzzy Systems*, pp. 1–6, AUG 2009.

[5] D. Metzler, Y. Bernstein, W. Croft, A. Moffat, and J. Zobel, "Similarity measures for tracking information flow," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, p. 524.

[6] V. Murdock, "Aspects of sentence retrieval," Ph.D. dissertation, University of Massachusetts Amherst, 2006.

[7] D. Metzler, S. Dumais, and C. Meek, "Similarity measures for short segments of text," *Lecture Notes in Computer Science*, vol. 4425, p. 16, 2007.

[8] R. Barzilay and N. Elhadad, "Sentence alignment for monolingual comparable corpora," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*, 2003, pp. 25–32.

[9] B. Dolan, C. Quirk, and C. Brockett, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics Morristown, NJ, USA, 2004.

[10] J. Cordeiro, G. Dias, and P. Brazdil, "A Metric for Paraphrase Detection," in *Computing in the Global Information Technology, 2007. ICCGI 2007. International Multi-Conference on*, 2007, pp. 7–7.

[11] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 775.

[12] I. Dagan, O. Glickman, and B. Magnini, "The pascal recognising textual entailment challenge," *Lecture Notes in Computer Science*, vol. 3944, pp. 177–190, 2006.

[13] A. Hickl and J. Bensley, "A discourse commitment-based framework for recognizing textual entailment," *ACL 2007*, p. 171.

[14] M. Tatu and D. Moldovan, "COGEX at RTE3," in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007, pp. 22–27.

[15] M. Bilotti, P. Ogilvie, J. Callan, and E. Nyberg, "Structured retrieval for question answering," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in information retrieval*. ACM New York, NY, USA, 2007, pp. 351–358.

[16] S. Shehata, F. Karray, and M. Kamel, "A concept-based model for enhancing text categorization," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, 2007, pp. 629–637.

[17] P. Achananuparp, X. Hu, and C. Yang, "Addressing the Variability of Natural Language Expression in Sentence Similarity with Semantic Structure of the Sentences," in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. Springer, 2009, p. 555.

[18] L. Kóczy, T. Vámos, and G. Biró, "Fuzzy signatures," in *EUROFUSE-SIC '99*, 1999, pp. 210–217.

[19] B. Mendis and T. Gedeon, "Aggregation selection for hierarchical fuzzy signatures: A comparison of hierarchical owa and wrao," in *IPMU'08*, 2008.

[20] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[21] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone," in *Proceedings of the 5th annual international conference on Systems documentation*. ACM New York, NY, USA, 1986, pp. 24–26.

[22] S. Banerjee, "Adapting the Lesk algorithm for word sense disambiguation to WordNet," Ph.D. dissertation, Citeseer, 2002.

[23] "Jaccard's similarity coefficient." [Online]. Available: http://en.wikipedia.org/wiki/Jaccard_similarity_coefficent

[24] D. Lin, "Using syntactic dependency as local context to resolve word sense ambiguity," in *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, vol. 35. ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 1997, pp. 64–71.

[25] P. Achananuparp, X. Hu, and X. Shen, "The evaluation of sentence similarity measures," in *DaWaK08: Proceedings of the 10th interna-*

tional conference on Data Warehousing and Knowledge Discovery. Springer, pp. 305–316.

[26] P. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Pearson Addison Wesley Boston, 2005.

[27] S. Ponzetto and M. Strube, "Knowledge derived from Wikipedia for computing semantic relatedness," *Journal of Artificial Intelligence Research*, vol. 30, no. 1, pp. 181–212, 2007.

[28] M. Sahami and T. Heilman, "A web-based kernel function for measuring the similarity of short text snippets," in *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006, p. 386.