Journal of Computer Science

Original Research Paper

An Approach for Finding Semantic Relatedness Score Between Two Sentences Based on their Senses

[1]Nazreena Rahman and [2]Bhogeswar Borah

[1]Department of Computer Science and Engineering, Kaziranga University, India
[2]Department of Computer Science and Engineering, Tezpur University, India

Article history
Received: 24-11-2020
Revised: 23-02-2021
Accepted: 24-02-2021

Corresponding Author:
Nazreena Rahman
Department of Computer
Science and Engineering,
Kaziranga University, India
Email: nazreena.rehman@gmail.com

Abstract: Finding semantic relatedness score between two sentences is useful in many research areas. Existing relatedness methods do not consider its sense while computing semantic relatedness score between two sentences. In this study, a Word Sense Disambiguation (WSD) method is proposed which is used in finding the sense-oriented sentence semantic relatedness measure. The WSD method is used to find the correct sense of a word present in a sentence. The proposed method uses both the WordNet lexical dictionary and the Wikipedia corpus. The sense-oriented sentence semantic relatedness measure combines edge-based score between words depending the context of the sentence; sense based score which finds sentences having similar senses; as well as word order score. We have evaluated the proposed WSD method on publicly available English WSD corpora. We have compared our proposed sense-oriented sentence semantic relatedness measure on standard datasets. Experimental analysis illustrates the significance of proposed method over many baseline and current systems like Lesk, UKB, IMS, Babelfy.

Keywords: Word Sense Disambiguation (WSD), Sense-Oriented Sentence Semantic Relatedness Measure, Semantic Relatedness Score, Edge-Based Feature, Sense Based Feature, Word Order Information

Introduction

In the era of Natural Language Processing (NLP), recent applications require an efficient process for calculating semantic relatedness score between two sentences, phrases or words. Semantic relatedness is a mea- sure of conceptual distance between two words based on their meaning. Relatedness measure helps in many applications like information retrieval, text mining, web page retrieval and dialog systems. Text mining tasks include text clustering, text summarization, text categorization, sentiment analysis. Semantic relatedness helps in finding unknown knowledge from textual databases (Atkinson-Abutridy *et al*., 2004). For the biomedical applications, semantic relatedness calculation is used as an important tool in gene expression, gene clustering and disease gene prioritization (Pesquita *et al*., 2009; Lord *et al*., 2003; Pedersen *et al*., 2007). Semantic relatedness score calculation between query and input text sentences is important in query-based text summarization. Semantic relatedness calculation is an emerging research topic which is widely used in many recent application fields (Hasan *et al*., 2020a; 2020b; Park *et al*., 2019; Zhu *et al*., 2019; Sadr *et al*., 2019).

Existing relatedness methods do not consider its sense while computing semantic relatedness score between two sentences. In this study, we introduce a sense-oriented semantic relatedness measure which incorporates sense of the sentence. Sense adds correct meaning to the sentence itself. Proposed method finds semantic relatedness score between two sentences based on specific sense present in WordNet.

The remainder of the paper is as follows: Section 2 presents the literature review on different techniques used in semantic similarity and relatedness measure. Section 3 states the motivation. Section 4 provides the brief introduction of the proposed technique. Section 5 describes the importance of sense for finding semantic relatedness measure. Section 6 presents the proposed unsupervised method for word sense disambiguation. Section 7 describes how to find the sense-oriented sentence semantic relatedness measure between two sentences in detail. Section 8 illustrates one example in detail. Section 9 includes the experimental analysis and discussion. Finally, this paper concludes with section 10 with some feasible future scopes.

Related Work

Different recent works are done in the area of natural language processing by providing different solutions for calculating semantic similarity between two text sentences. In this section, different related works will be discussed to explore the advantages and disadvantages of existing methods. Related works can be classified into following categories:

- Semantic similarity based on vector space model
- Semantic similarity based on ontology
- Web search engine based measure

Semantic Similarity Based on Vector Space Model

The vector space model is an algebraic model. It has two steps, first step is to convert the text documents into vector of words and second step is to convert it to a numerical form to apply text mining techniques. It is widely used in information retrieval system (Meadow *et al*., 1992). In information retrieval, there is a compiled word list having *n* words. This word list includes all the meaningful words present in natural language. Each document is represented as a vector in n-dimensional space. Query is associated with each document and also is represented as a vector. To compute these values, tf-idf weighting scheme is widely used (Hiemstra, 2000). Necessary documents are retrieved based on similarity between document vector and query vector (Pawar and Mago, 2018). Similarity will be high if both vectors have same words. The advantages of this method are:

- It can extract keywords from the documents
- It matches with documents, hence does not depend on size of the document

Although this retrieval mechanism is quite simple, but it has some drawbacks:

- It does not consider order of the words present in a sentence
- It does not consider semantic relatedness between two sentences
- A same word might have different meaning for different sentences. Therefore, sense should also be taken into account

Semantic Similarity Based on Ontology

A lexical dictionary is used here to compute the semantic similarity between word pairs present in each sentence. We have to combine the word level similarity to get sentence level similarity. Sentence similarity is computed by aggregating semantic similarity values of all word pairs (Okazaki *et al*., 2003). The word 'relatedness' or 'similarity' is defined as the number of shared common features of meaning. Relations be- tween two words can be defined on the basis of thesaurus methods or distributional methods. Thesaurus based method normally use on-line thesaurus; example: WordNet, MeSH. Distributional methods depends on the distribution of two words in a corpus. There are mainly four different thesaurus (Gupta and Yadav, 2014) based methods:

- Structure or Edge based Measures
- Information Content Measures
- Feature-Based Measures
- Hybrid Measures

Structure or Edge Based Measures

In case of structure or edge based semantic similarity measure, it finds the path length in ontology hierarchy structure. It computes the length of the path linking the words and on the position of the words in the ontology. Similarity between two words increases if they have more links among the concepts (Rada *et al*., 1989; Richardson *et al*., 1994):

- Shortest path measures: Shortest path measure finds the closeness of two words in hierarchical semantic nets (Rada *et al*., 1989). Less similar words have greater distance. with less shared common features of meaning. Let *W*1 and *W*2 are two words, the equation to find shortest path is:

  SP (*W*1, *W*2) = Number of edges in the shortest path

- Wu and Palmer Measure: This similarity measure finds the most specific common ancestor of *W*1 and *W*2. The most specific common ancestor is known as Least Common Sub-summer (LCS). Figure 1 shows that for the word *Minicab* and *Gypsy_cab*, the LCS is *cab*
  Wu and Palmer measure depends on depth of the two words in the taxonomy hierarchy. It also considers the depth of the least common sub-summer (Wu and Palmer, 1994). This measure can be calculated:

$$Wu\,\&\,P(W1,W2) = \frac{2\times depth(LCS(W1,W2))}{depth(W1)+depth(W2)+2\times depth(LCS(W1,W2))}$$

depth (*LCS*(*W*1, *W*2)) = Number of edges from the least common sub-sum of *W*1 and *W*2 to the root node *depth* $(W1)$ = Number of edges from W1 node to the least common sub-sum of W1 and W2

depth $(W2)$ = Number of edges from W2 node to the least common sub-sum of W1 and W2

- Leacock-Chodorow Measure: Leacock-Chodorow gives the following similarity measure Leacock and Chodorow (1998):

$$Le \,\&\, Ch(W1,W2) = -\log \frac{shortest\ path(W1,W2)}{2\times D}$$

Here, *shortest path*($W1$, $W2$) defines the shortest path length between two words $W1$ and $W2$ in the taxonomy hierarchy and D is the maximum depth of the taxonomy

Information Content Measures

Information content based measures depend on amount of common information shared between two words. If two words share more information in common, then the information content increases. In Information Content (IC), it finds the probability of the word W ($P(W)$). IC of W is calculated by counting the frequency of that word in the corpus divided by the total number of words present in the corpus:

$$IC(W) = -\log(P(W))$$

We will discuss a few information content based semantic similarity measures in this following section:

- Resnik Similarity Measure: Resnik similarity (Resnik, 1995) directly depends on the commonly shared in- formation. Shared information is carried out by finding information content of least common subsume of two words W1 and W2 in the taxonomy hierarchy:

$$Resnik(W1,W2) = IC(LCS(W1,W2))$$

Here, LCS stands for least common sub-sum.

- Jiang and Conrath Similarity Measure: (Jiang and Conrath, 1997) calculates the similarity measure by considering the information content of two words along with information content of their least common subsume:

$$Jiang \,\&\, ConDistance(W1,W2) = \\ IC(W1) + IC(W2) - 2\times IC(LCS(W1,W2))$$

This Jiang and ConDistance measure gives dissimilarity between two words. As the dis-similar value increases, it shows low relatedness as well as low similarity between two words.

- Lin Similarity Measure: In case of Lin similarity (Li *et al*., 2003), it captures the similarity between two words as the ratio of amount of information shared between two words to the information possessed by each word:

$$LinSimilarity(W1,W2) = \frac{2\times IC(LCS(W1,W2))}{IC(W1)+IC(W2)}$$

Feature-Based Measures

Based on the description of the feature of words, feature-based similarity measures are introduced. Feature of a word contains the important information about the word. Each word is described in terms of set of words (definitions or glosses in WordNet) that contains the properties. Feature based similarity methods try to over-come the disadvantages of path based similarity method in the fact that taxonomical links present in a ontology do not necessarily give uniform distances. Here, each term is described by set of words which indicates properties or features.

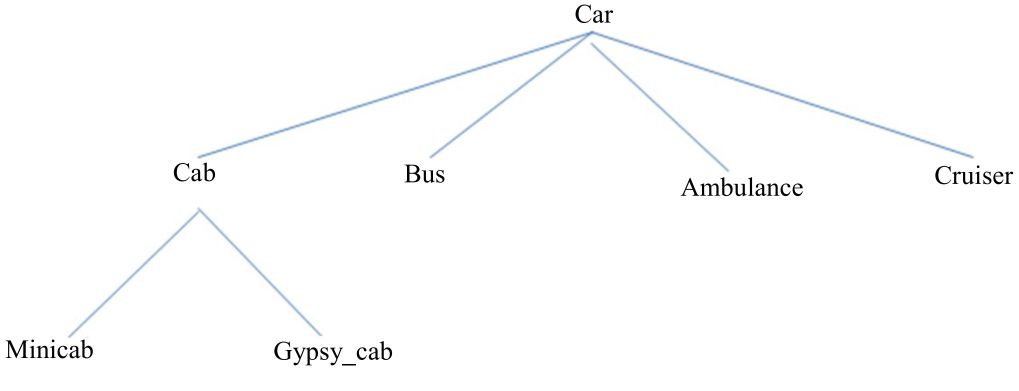

Fig. 1: An example of Least Common Sub-sumer (LCS) in WordNet hierarchy

Similarity gets increased if two words share more common features. They can find similarity between two words if they are present in two different ontologies (Slimani, 2013):

- Tversky's similarity Measure: Based on the description of the two words *W*1 and *W*2, (Tversky, 1977) has put forward a feature based similarity measure. It basically calculates the common words between the description of the *W*1 and *W*2, words present in *W*1 but not in *W*2 and words present in *W*2 but not in *W*1:

$$TverskySimilarity\left(W1,\ W2\right) = \frac{\left|W1 \cap W2\right|}{\left|W1 \cap W2\right| + \alpha\left|\left(W1 - W2\right)\right| + \beta\left|\left(W2 - W1\right)\right|}$$

where, α, $\beta \geq$_ and $\alpha + \beta = 1$

- Rodriguez and Egenhofer Similarity Measure: Rodriguez and Egenhofer (2003) computes similarity between two words based on their synsets, definition and neighboring words. They calculate the similarity between two words as the weighted sum of linear equation of all three:

$$Ro\ \&\ Eg\ Similarity\left(W1,W2\right) = u \cdot Ssynsets\left(W1,W2\right) + v \cdot Sfeatures\left(W1,W2\right) + w \cdot Sneighborhoods\left(W1,W2\right)$$

u, *v* and *w* are these three weighting parameters. They depend on characteristics of ontology. *S* is the overlapping of different features calculated as:

$$S\left(W1,W2\right) = \frac{\left|w1, w2\right|}{\left|w1 \cap w2\right| + A\left|\left(w1 \setminus w2\right)\right| + \left(1 - A\right)\left|\left(w2 \setminus w1\right)\right|}$$

Here:

$$\gamma\left(W1,W2\right) \ = \ A$$

*w*1 and *w*2 are the definitions for the words corresponding to *W*1 and *W*2. *w*1\w2 defines the set of words present in *w*1 but not in *w*2 and *w*2\w1 stands to represent the set of words present in *w*2 but not in *w*1. Finally, γ(*W*1, *W*2) is a function that depicts the depth of *W*1 and *W*2 in the taxonomy:

$$\gamma\left(W1,W2\right) \ = \ \begin{cases} \dfrac{depth\left(W1\right)}{depth\left(W1\right) + depth\left(W2\right)} \\ if\ depth\left(W1\right) \leq depth\left(W2\right) \\ \dfrac{depth\left(W2\right)}{depth\left(W1\right) + depth\left(W2\right)} \\ if\ depth\left(W1\right) > depth\left(W2\right) \end{cases}$$

- X-Similarity Measure: A feature based function is proposed by (Petrakis *et al*., 2006) which is known as X-similarity measure. According to them, two words are similar if the synsets and glosses of their words and their neighborhood words are lexically similar. In this similarity, it finds the matching between synsets and gloss of two words and their neighborhood words extracted fromWord-Net:

$$x - similarity\left(W1,W2\right) = \begin{cases} 1 \\ if\ S_{synsets}\left(W1,W2\right) > 0 \\ \max\left(S_{neigh}\left(W1,W2\right), \right. \\ \left. S_{glosses}\left(W1,W2\right)\right) \\ if\ S_{synsets}\left(W1,W2\right) = 0 \end{cases}$$

Sneigh(*W*1, *W*2) can be calculated as follows:

$$S_{neigh}\left(W1,W2\right) = \max \lim_{i \in SR} \frac{\left|wl_i \cap wl_2\right|}{\left|wl_i \cup wl_2\right|}$$

Each different Semantic Relation type (SR) is considered (such as *is-a* and *part-of* inWordNet) and take the highest value among them. $S_{synsets}$(*W*1, *W*2) and $S_{glosses}$(*W*1, *W*2) both can be calculated as:

$$S\left(W1,W2\right) = \frac{wl \cap w2}{wl \cup w2}$$

where, *w*1 and *w*2 are the set of synsets and glosses for the words *W*1 and *W*2.

Hybrid Measures

In hybrid measure, it combines the characteristics of different similarity measures as it is described above.

This combination gets higher accuracy in many similarity measures:

- Knappe Similarity Measure: This measure (Knappe *et al*., 2003) based on the fact that there are multiple paths to link two words in the taxonomy hierarchy. The proposed measure is given as:

$$S_{knappe}\left(W1,W2\right) = p \times \frac{\left|Ans_{W1} \cap Ans_{W2}\right|}{Ans_{W1}} + \left(1 - p\right) \times \frac{\left|Ans_{W1} \cap Ans_{W2}\right|}{Ans_{W2}}$$

where, value of *p* is in between 0 and 1. *AnsW*1 and *AnsW*2 both defines the ancestor nodes of the words *W*1 and *W*2 respectively. The reachable nodes shared by both *W*1 and *W*2 is $Ans_{W1} \cap Ans_{W2}$.

- Zhou Similarity Measure: Both path based and information content based measures are integrated by (Zhou *et al*., 2008) with the help of the following equations:

$$S_{zhou}\left(W1,W2\right)=1-k\left(\frac{\ln\left(len\left(W1,W2\right)+1\right)}{\ln\left(2\times\left(deep_{max}-1\right)\right)}\right)$$

$$\left(1-k\right)\times\left(\left(IC\left(W1\right)+IC\left(W2\right)-2\times IC\left(LCS\left(W1,W2\right)\right)/2\right)\right.$$

Here, $LCS(W1, W2)$ stands for least common sub-sum of $W1$ and $W2$. If the value of $k = 1$, then the Zhou Similarity Measure will be a path based measure if $k = 0$, then it will be an information content based measure. $deep_{\max}$ is the depth of the taxonomy

Web Search Engine Based Measure

This methodology computes relatedness based on web search engine results. It finds the frequently occurring words together (Bollegala *et al*., 2007). All the above mentioned applications are domain specific and use different methods to get the results.

Motivation

From the literature survey, it is seen that many different techniques have been applied in semantic similarity/relatedness measure till now, but sense of a sentence is not incorporated while finding similarity or related- ness score between two text sentences. It is also seen that, no semantic relatedness or similarity measure is there in which score can be found on sentence level. A word should take that sense which is suitable with context of the sentence. Existing semantic measures depend on individual words. Therefore, it is important to find exact sense of both words for which the senses are appropriate for the sentences. We have added sense as a measure while finding semantic relatedness measure. We have also added word order similarity between two sentences by finding its longest common substring. Finally, semantic relatedness measure, sense relatedness measure and word order measure are added in an equation to get proper sense-oriented semantic relatedness score between two sentences.

Proposed Method for Finding Sense-Oriented Sentence Semantic Relatedness Measure

Sense-oriented sentence semantic relatedness score between two sentences is calculated by combining the semantic relatedness score, sense relatedness score and word order similarity score. The process for finding semantic relatedness score, sense relatedness score and word order similarity score between two sentences is shown in the following Fig. 2 to 4.

These following steps are described briefly for finding semantic relatedness between two sentences:

1. Pre-processing: Initially, pre-processing is done on text sentences by applying various techniques proposed by linguists. Steps for sentence pre-processing is already described in section 6
2. Word sense disambiguation on content word: To get proper semantic relatedness score between two sentences, appropriate sense of each word of the two sentences is disambiguated based on the context of the sentence. This word sense disambiguation technique is based on collocation score. The algorithm for extraction of proper sense to disambiguate a word is already described in section 6
3. Semantic relatedness score between two sentences: To get semantic relatedness score between two sentences, a lexical dictionary ‘WordNet’ is used. The detailed description of finding semantic relatedness score between two sentences is described in section 7.1
4. Sense based relatedness score between two sentences: To find the relatedness between two sentences based on their sense, sense based relatedness score is calculated in section 7.2
5. Word order score between two sentences: Order of words present in sentences also makes an impact while finding relatedness score between two sentences. The detailed description of word order similarity measure between two sentences is in section 7.3
6. Find sense-oriented semantic relatedness score between two sentences: To find out the sense-oriented sentence semantic relatedness score, all the three measures are added in a equation described in section 7.4

Importance of Sense for Finding Semantic Relatedness Measure

Query based text summarizer extracts semantically query related sentences from input text sentences. In most cases, to find the semantic relatedness score between two words using WordNet, existing measures find the score with all the senses and give maximum score. In WordNet, a word has many senses. For different types of part of speech of a content word, we get different senses. Senses are the gloss or the definition. For a content word, if it has more than one sense then different number senses have different glosses. A content word may contain different senses for a same part of speech. Table 1 shows about different gloss of word interest present in WordNet. Each synset of interest contains its parts-of-speech and sense number. Gloss implies a dictionary-style definition.

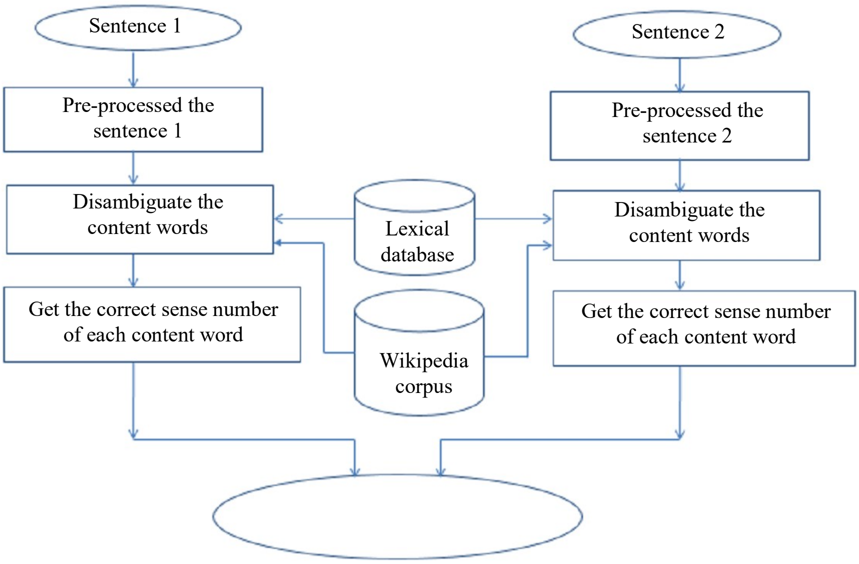
Sentence 1
Sentence 2
Pre-processed the sentence 1
Pre-processed the sentence 2
Disambiguate the content words
Lexical database
Disambiguate the content words
Get the correct sense number of each content word
Wikipedia corpus
Get the correct sense number of each content word

Fig. 2: Proposed steps to find semantic relatedness score between two sentences

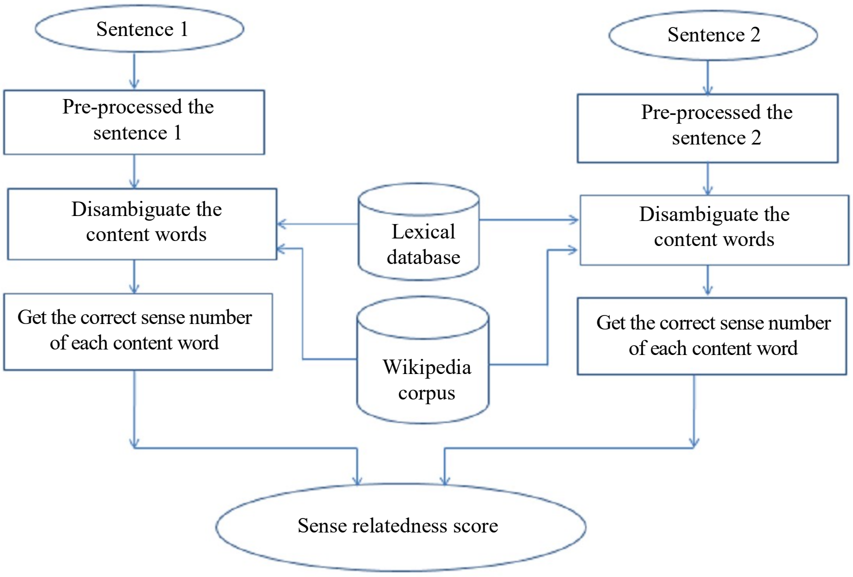
Sentence 1
Sentence 2
Pre-processed the sentence 1
Pre-processed the sentence 2
Disambiguate the content words
Lexical database
Disambiguate the content words
Get the correct sense number of each content word
Wikipedia corpus
Get the correct sense number of each content word
Sense relatedness score

Fig. 3: Proposed steps to find sense relatedness score between two sentences

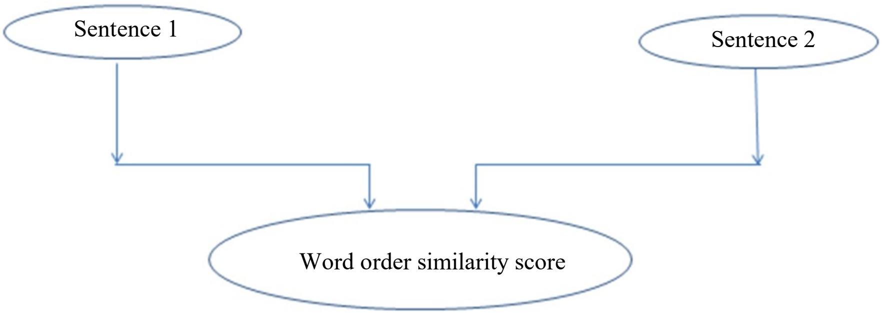

Fig. 4: Proposed steps to find word order similarity score between two sentences

One of the main issue in NLP is that presence of lexical ambiguity. Lexical ambiguity is a writing error that occurs when a sentence contains a word that holds more than one meaning. For example: We take two sentences: (1) Ram went to the bank to deposit money and (2) Ram went to the bank of river Brahmaputra. Here, the word bank has two different meaning; one is related with financial institute and other one is related with sloping land. Now, we find the semantic relatedness score between the content words of two sentences. In this example: We take two words that are *bank* and *river* where: *bank* word comes from the first sentence and *river* word comes from the second sentence. Both *bank* and *river* words are noun here. While finding the semantic similarity score, we have to give the word, then it's part of speech and sense number. When we do not give any particular sense as an input, WordNet takes automatically that sense for which it gets the highest semantic similarity score. UsingWordNet lexical dictionary, we get the following semantic relatedness scores for different measures listed in Table 2.

Table 2 shows that by default almost all semantic relatedness measures take the first sense of bank as it gives maximums score with river. Table 3 provides the different senses present for the word bank. For the word river only one noun sense is present in the WordNet described in Table 4.

Table 5 shows the trace definition present for *bank#n#1*, *bank#n#2*, *bank#n#3* and *river#n#1*. Trace definition shows how the word is present in WordNet taxonomy. From these tables, it is quite clear that though the word bank is actually related with the financial institution, here, by default all semantic relatedness measures take an incorrect sense of bank. Therefore, finding sense of a word is much essential to get accurate relatedness score between two words as well as between two sentences and will help in eliminating lexical ambiguity problem.

Table 1: Synset and gloss of word 'interest' in WordNet

Synset('interest.n.01')	A sense of concern with and curiosity about someone or something
Synset('sake.n.01')	A reason for wanting something done
Synset('interest.n.03')	The power of attracting or holding one's attention (because it is unusual or exciting etc.)
Synset('interest.n.04')	A fixed charge for borrowing money; usually a percentage of the amount borrowed
Synset('interest.n.05')	(law) a right or legal share of something; a financial involvement with something
Synset('interest.n.06')	(usually plural) a social group whose members control some field of activity and who have common aims
Synset('pastime.n.01')	A diversion that occupies one's time thoughts (usually pleasantly)
Synset('interest.v.01')	Excite the curiosity of; engage the interest of
Synset('concern.v.02')	Be on the mind of
Synset('matter to.v.01')	Be of importance or consequence

Table 2: Relatedness/similarity score between 'bank' and 'river'

Different semantic relatedness/similarity measure	Relatedness score
The relatedness of bank#n#1 and river#n#1 using vector pairs Li *et al.* (2009)	0.0353
The relatedness of bank#n#3 and river#n#1 using vector Li *et al.* (2009)	0.1958
The relatedness of bank#n#1 and river#n#1 using hso Hirst and St-Onge (1998)	0.0000
The relatedness of bank#n#1 and river#n#1 using Adapted lesk Banerjee and Pedersen (2002)	16.0000
The relatedness of bank#n#1 and river#n#1 using res Resnik (1995)	0.6144
The relatedness of bank#n#1 and river#n#1 using lch Leacock and Chodorow (1998)	1.4917
The relatedness of bank#n#1 and river#n#1 using lin Li *et al.* (2003)	0.0782
The relatedness of bank#n#1 and river#n#1 using jcn Jiang and Conrath (1997)	0.0691
The relatedness of bank#n#1 and river#n#1 using wup Wu and Palmer (1994)	0.4286
The relatedness of bank#n#1 and river#n#1 using path Rada *et al.* (1989)	0.1111

Table 3: Different senses present for the word 'bank' present in WordNet

Sense number	Meaning
1	Sloping land
2	A financial institution that accepts deposits and channels the money into lending activities
3	A long ridge or pile
4	An arrangement of similar objects in a row or in tiers
5	A supply or stock held in reserve for future use
6	The funds held by a gambling house or the dealer in some gambling games
7	A slope in the turn of a road or track
8	A container (usually with a slot in the top) for keeping money at home
9	A building in which the business of banking transacted
10	A flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)

Table 4: Sense present for the word 'river'

Sense number	Meaning
1	A large natural stream of water (larger than a creek)

Table 5: Trace Definition present in WordNet

Concept	Trace definition
bank#n#1	*Root*#n#1 entity#n#1 physical entity#n#1 object#n#1 geological formation#n#1 slope#n#1 bank#n#1
bank#n#2	*Root*#n#1 entity#n#1 abstraction#n#6 group#n#1 social group#n#1 organization#n#1 institution#n#1 financial institution#n#1 depository financial institution#n#1
bank#n#3	*Root*#n#1 entity#n#1 physical entity#n#1 object#n#1 geological formation#n#1 natural elevation#n#1 ridge#n#1 bank#n#3
river#n#1	*Root*#n#1 entity#n#1 physical entity#n#1 thing#n#12 body of water#n#1 stream#n#1 river#n#1

Proposed Unsupervised Method for Word Sense Disambiguation

The overall process for finding the sense of a target word is shown in the following Fig. 5.

Following steps are described briefly for finding sense of a word present in a text sentence using collocation score:

- Pre-processing: Initially, pre-processing is done to remove the unwanted words from the text sentence. Here, the unwanted words mean stop words and person, location or organization's name. This makes the text sentence a lighter one. Following techniques are applied to do the pre-processing of text document:
    - Part of Speech Tagging: To classify the words on the basis of part of speech category, part of speech tagging (Bird *et al*., 2009) is done. part of speech tagging classifies the content words. Tags include noun, adjective, verb and adverb
    - Named Entity Tagging: To distinguish different Names person, location or organization names, we do the named entity tagging (Bird *et al*., 2009). We will not consider person's name to find semantic relatedness as it is not present in Lexical resources
    - Stop Word Removal: It is better to filter words like out a, an, the, in etc. which do not give any semantic meaning to the sentence. This is known as stop word removal in text mining applications. Here, we use stop word list stores in Natural Language Tool Kit (NLTK) in python
    - Stemming: Finally, stemming is done to the content words. Stemming brings the word to its root or base form. For example to convert a word from plural to singular root form (girls to girl) or removing *ing* from a verb (singing to sing). Number of algorithms are available to do the stemming in natural language processing
- To get the correct sense of a word, we take collocation feature. Following section shows how collocation feature can be applied for finding word sense

Finding Collocation Score between Two Words

Collocation refers to the word or phrase that is often used with other word or phrase. With collocation, we can find what words occur near other words. The computational technique that finds commonly collocated words or phrases in a document or corpus is known as collocation extraction. Collocation score between two words is calculated by finding the number of occurrence of those words together in a corpus. Here, Wikipedia corpus (Denoyer and Gallinari, 2006) is used. The co-occurrence between two terms is calculated by finding its bi-gram frequency. Collocation gives the associativity between two words. For example: Car and bike are two concepts or words that are semantically similar. They have some common features like wheels or have common function like transport. In contrast, car and petrol both are associated as they occur frequently in language and space. This can be said as functional relationship. Association and similarity both are neither mutually exclusive nor independent. Car and patrol both are related two both relations to some degree (McRae *et al*., 2012; Plaut, 1995). To find the bi-gram collocation score for each sense of word w_1 (McKeown and Radev, 2000), we find frequency of occurrence of words present in the sense definition for w_1 with other words present in the sentences and take the maximum one. For example, we take one sentence Mary treated John for his injuries. To find the exact sense of the word treat, we first find out the all the senses present for treat in WordNet. Senses are the glosses or the definitions. The method finds the collocation score of each word present in the gloss with the word present in the sentence. Here, one gloss for treat is interact in a certain way. The content words present in the sentence is injuries and in the definitions are interact, certain, way. After finding collocation score of interact, certain, way with injuries, the method takes the highest score. In this way method will calculate for each sense and finally we take that sense for which the collocation score is maximum.

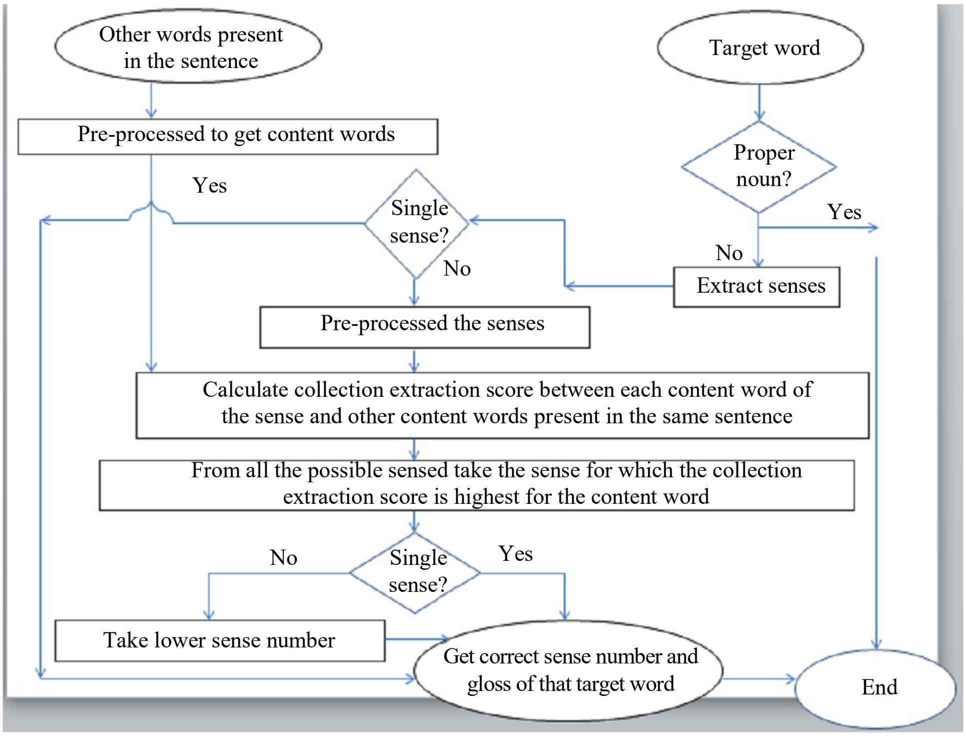
Other words present in the sentence
Target word
Pre-processed to get content words
Proper noun?
Yes
Single sense?
Yes
No
No
Extract senses
Pre-processed the senses
Calculate collection extraction score between each content word of the sense and other content words present in the same sentence
From all the possible sensed take the sense for which the collection extraction score is highest for the content word
No
Single sense?
Yes
Take lower sense number
Get correct sense number and gloss of that target word
End

Fig. 5: Block diagram of word sense detection method

The proposed WSD method will work for all the content words present in WordNet except the person's name. We will not consider the person's name, but of course, we will include an organization or location's name using standard Named Entity Tagger (Perkins, 2014). The proposed method considers mainly the content words as they carry the salient information. Content word includes noun, main verb, adjective and adverb. First, the proposed WSD method finds all the senses of the target words present in the WordNet. Senses are the glosses. For each sense of a target word, we have removed the stop words. We also remove the stop words from the sentence in where target word is present. To find the collocation score between two words (one word is from the gloss of *w* and other word is from the sentence *w*'), we use the following Equation 1. Here, Wikipedia corpus is used (Denoyer and Gallinari, 2006):

$$collocation\_score(w,w') = \frac{\log\left(\frac{x * sizeCorpus}{(w * w' * span)}\right)}{\log(2)} \quad (1)$$

Where:

w = Frequency of the word w present in the Wikipedia corpus
w' = Frequency of the word w' present in the Wikipedia corpus
x = Frequency of w' near w present in the Wikipedia corpus
sizeCorpus = Size of the Wikipedia corpus
span = Width of words (e.g., 3 to left and 3 to right of first word)

While finding the collocation extraction score, we are giving the flexibility that if the two words are not together in Wikipedia, we increase the window size up-to 3. We consider the span size as 3 because it works best for our proposed WSD method. We will search for bi-gram frequency where the words may be separated by three other words in the text of Wikipedia. To find the sense of a word present in a sentence, initially, we get a set of probable senses of the word. Now for each sense, we calculate the collocation score between each content word of every gloss of a sense with all other content words present in the same sentence. Same process will be followed for every sense and finally we take that sense for which the collocation score is maximum. The Collocation Score (CS) of a sense (gloss) for a Target Word (TW) present in the sentence is:

$$CS(Scense, Sentence) = \max \sum_{w \in Sense, w' \in Sentence} (collocation\,score(w,w')) \quad (2)$$

After finding the collocation score of the set of all senses of *TW*, we consider that sense of *TW* for which the collocation score is maximum.

Finding Exact Sense of a Word Present in a Sentence

The proposed WSD method is implemented to find the sense of a word which will further helps in calculating semantic relatedness score between query and input text sentences. Following Algorithm 1 gives the systematic steps to find the sense of a word:

Algorithm 1: Steps to Find Exact Sense of a Word present in the Sentence

```
     Data: Target Word ($T$) and Other Words Present in
           the Sentence ($OT$)
     Result: Sense Number and Gloss of that Sense of $T$ for
             Which the Collocation Score is Maximum
1    Do the part of speech of T
2    if T is a person's name then
3        Go to step 24
4    end
5    else
6        Find out the senses of $T$ using WordNet where
7        if $T$ has more than one sense then
8            for each sense ($s$) of $T$ do
9                Do the pre-processing of $s$
10               Do the pre-processing of $OT$
11               Find out the collocation score between s
                 and $OT$ by using Equation 2
12           end
13           Extract the sense number and the gloss of $T$ for
             which we get the maximum collocation score
14           if $T$ has more than one sense having same
             maximum collocation score then
15               Extract the sense number and gloss which
                 has lower sense number
16           end
17           else
18               Go to step 21
19           end
20           else
21               Get the sense number and the gloss
22           end
23       end
24   end
```

Detailed Description of the Sense-Oriented Semantic Relatedness Calculation between Two Sentences

Relatedness implies larger set of potential relationship between words. Antonyms are highly related words but they are less similar. Human and women are not similar but highly related than human and car while girl and women have similarity. Thus Similarity is a sub case of Relatedness. We prefer to use semantic relatedness score to find query relevant sentences. It is quite clear that semantic measures needs actual sense while finding its score between two words. A word's sense depends on other words present in the same sentence. In case of semantic relatedness calculation, default sense does not give the assurance of accuracy of score for specific sentences. Hence, sense detection is important to increase accuracy of measures. To find sense-oriented semantic relatedness score we combine sentence semantic relatedness score, sense relatedness score and word order relatedness score. In the following section, sense-oriented semantic relatedness score calculation is shown sequentially.

Finding Semantic Relatedness Score between Two Sentences

To find the relatedness between two sentences, semantic relatedness score is calculated between two words present in both sentences. Semantic relatedness mea- sure gives the relatedness score between two sentences on the basis of the meaning of the sentences. This relatedness measure is based on word to word relatedness.

We consider only the content words while calculating the semantic relatedness score between two sentences. Initially, pre-processing step removes the stop words. Stemming is also done to get the root form of a word. Before finding relatedness score between two words, it is important to find exact sense of that word. Full description of the word sense disambiguation method is described in Algorithm 1.

This relatedness score uses Path Weight measure (Pedersen *et al*., 2004; Hirst and St-Onge, 1998). It is a semantic relatedness measure between two words based on path which is described in WordNet lexical dictionary. It is found from the literature survey that this measure includes more relations and can find relatedness score between different part of speech (Patwardhan *et al*., 2003). It classifies the relations in WordNet as up-ward, downward or horizontal directions. Higher score implies the shorter path length and less changes of directions. While finding the semantic relatedness score between two words, the proposed method will give the accurate sense number and its part of speech along with the respective words. The equation to find Semantic Relatedness Score (*SEM_R_S*) between two words $w1$ and $w2$ is:

$$\begin{aligned}&SEM\_R\_S\big((w1, sense\_no\_1, p\_o\_s\_1),\\&(w2, sense\_no\_2, p\_o\_s\_2)\big) = 2\times c -\\&PathLength(w1,w2) - k\times DirectionChange(w1,w2)\end{aligned} \tag{3}$$

Here:

$sense\_no\_1$	=	Sense number of $W1$
$p\_o\_s\_1$	=	Part of speech of $W1$
$sense\_no\_2$	=	Sense number of $W2$
$p\_o\_s\_2$	=	Part of speech of $W2$

Here, c and k are the constants and values are $c = 8$ and $k = 1$. The maximum semantic relatedness score between two word is 16 which signifies that two content words are identical. The minimum score is 0 which signifies there is no semantic relatedness between them. Following equation ?? is used to find the Semantic Relatedness Score (*SEM_R_S*) between two sentences $s1$ and $s2$:

$$SEM\_R\_S(s1,s2) = \sqrt{\frac{\sum_{w1\in s1,w2\in s2} SEM\_R\_S((w1,sense\_no\_1,p\_o\_s\_1) + (w2,sense\_no\_2,p\_o\_s\_2)}{Maximum\ relatedness\ score}} \tag{4}$$

The overall process to find semantic relatedness score between two sentences is shown in the Algorithm 2.

Algorithm 2: Steps to find semantic relatedness score between two sentences

```
Data: Two Sentences ($s1$, $s2$)
Result: Semantic Relatedness Score between $s1$ and $s2$ ($SEM\_R\_S(s1, s2)$)
Do the pre-processing of $s1$, $s2$ by using the steps mentioned in section 6
for each content word $w1$ in $s1$ and $w2$ in $s2$ do
    Find out the sense number of $w1$ and $w2$ by using the Algorithm 1
end
for each content word $w1$ and $w2$ from $s1$ and $s2$ do
    Find out the $SEM\_R\_S$ (($w1$, sense_no_1, p_o_s_1), ($w2$, sense_no_2, p_o_s_2)) by using the Equation 3
end
Find the $SEM\_R\_S(s1, s2)$ between $s1$ and $s2$ by using the Equation 4
```

Finding Sense Relatedness Score between Two Sentences

We have already discussed that finding relatedness measure between two sentences on the basis of its sense is quite important. Proposed WSD method finds the sense of each content word present in both text sentences by using the mentioned method described in Algorithm 1. After doing the pre-processing of two sentences as mentioned in section 6, we have disambiguated the sense of each content word if it has more than one sense. We get the gloss for each content word. We will do the stop word removal and stemming on the content words to get the root form of the content words present in the gloss. Here, we find the sense relatedness score between two sentences by finding the common content words present in the gloss of a sense of a content word in the first sentence with the words present in the second sentence. The method uses the Equations 5 to 7 to find the Sense Relatedness Score (*S_R_S*) between two sentences s_1 and s_2:

$$S\_R\_S\_1(s_1,s_2) = \max \sum_{w\in s_1} \{Sense\ definition\ of\ w \cap Words\ present\ in\ s_2\} \tag{5}$$

$$S\_R\_S\_2(s_1,s_2) = \max \sum_{w'\in s_2} \{Sense\ definition\ of\ w' \cap Words\ present\ in\ s_1\} \tag{6}$$

$$S\_R\_S(s_1,s_2) = \frac{S\_R\_S\_1(s_1,s_2) + S\_R\_S\_2(s_1,s_2)}{Total\ number\ of\ content\ words\ present\ in\ s_1\ and\ s_2} \tag{7}$$

The overall process to find the sense relatedness score between two sentences is shown in the Algorithm 3.

Algorithm 3: Steps to Find Sense Relatedness Score between Two Sentences

```
Data: Two Sentences (s1; s2)
Result: Sense Relatedness Score between $s1$ and $s2$ ($S\_R\_S(s1, s2)$)
Do the pre-processing of $s1$ and $s2$ by using the steps mentioned in section 6
Find out the gloss of the sense of the content words by using the Algorithm 1
Do the pre-processing of the gloss of the sense of the content words by using the steps mentioned in section 6
Find the $S\_R\_S(s1, s2)$ between $s1$ and $s2$ by using the Equation 7
```

Finding Word Order Similarity Score between Two Sentences

On the basis of same sequence of words present in the two sentences, word order similarity provides how much two sentences are similar. Finding longest common substring between two sentences adds more impact on similarity measure. Word order similarity method depends on the order of words present in both sentences. It helps in signifying the relatedness between two sentences though they share same words. Example: (a) Ram killed Shyam and (b) Shyam killed Ram. These two sentences use same content words but we can see that sentence *a* and sentence *b* have opposite meaning. Longest common substring can easily distinguish that meaning of sentence *a* is not similar to the meaning of sentence *b*. We can take another example: Sentence 1 is Narendra Modi's visit to China and sentence 2 is Ram Nath Kovind's visit to China. Both sentences are different though they carry maximum common words. Hence finding longest common substring can identify the differences present in both sentences. It can also find similarity between two sentences if numerical data present in both sentences in a same order. Ontology based semantic relatedness measure can not find this type of similarity as lexical dictionary does not contain numerical data or some proper nouns. Example: Sentence 1: In 2006, Ram came to Assam to meet his friend Rahim and sentence 2: In 2006, Ram came to Guwahati to meet Rahim. Here longest common substring is in 2006, Ram came to. Hence, it also helps in finding similarity for numerical and proper noun words. Here, we will not do any pre-processing task. The method uses the following equation ?? to find word order similarity between two sentences $s1$ and $s2$:

$$W\_O\_S(s_1,s_2)=\frac{NCW(s_1 s_2)}{TNWLS(s_1 s_2)} \tag{8}$$

Here:

NCW = Number of Common Words between s_1 and s_2
TNWLS = Total Number of Words present in the Longest Sentence between s_1 and s_2

Finding Sense-Oriented Sentence Semantic Relatedness Score between Two Sentences

Semantic relatedness measure gives how much two sentences are related to each other. Sense relatedness measure defines how much two sentences are related on the basis of its sense definition and word order similarity gives information relations between sentences. All these semantic and same sequence of words information play an important role while finding sense-oriented sentence semantic relatedness score between two sentences. Hence, the sense-oriented sentence semantic relatedness measure between the two sentences is the combination of all these three measures: Semantic relatedness, sense relatedness and word order similarity.

The sense-oriented sentence semantic relatedness score between sentences s_1 and s_2 is given in Equation 9:

$$Sense\_Sem\_Rel(s_1,s_2)=\alpha\times SEM\_R\_S(s1,s2) \\ +\beta\times S\_R\_S(s_1,s_2)+\gamma\times W\_O\_S(s_1,s_2) \tag{9}$$

Here $\alpha + \beta + \gamma = 1$ and α, β and are weighting parameters. They specify relative contributions to the sense-oriented sentence semantic relatedness measure of semantic, sense and word order measures. As semantic information caries more relevant information, therefore more weightage is given to semantic relatedness measure (Wiemer-Hastings, 2000). Considering the view that the word order information plays a subordinate role in finding relatedness between sentences, hence the weightage given to word order information is minimum. The overall process to find sense-oriented sentence semantic relatedness score between two sentences is shown in the Algorithm 4.

Algorithm 4: Steps to find sense-oriented sentence semantic relatedness score

```
Data: Two Sentences (s1; s2)
Result: Sense-Oriented Sentence Semantic Relatedness
        Score (Sense_Sem_Rel) of $s1$, $s2$
1 Find semantic relatedness score between $s1$ and $s2$ by
  using the Algorithm 2
2 Find sense relatedness score between $s1$ and $s2$ by
  using the Algorithm 3
3 Find word order similarity score between $s1$ and $s2$ by
  using the Equation 8
4 Find sense-oriented sentence semantic relatedness
  score between $s1$ and $s2$ by using Equation 9
```

Implementation of the Proposed Sense-Oriented Sentence Semantic Relatedness Score Finding Method

To illustrate the implementation of the sense-oriented sentence semantic relatedness measure between two sentences, we have elaborated the proposed method with three sentences $S1$, $S2$ and $S3$:

- Mary treated John for his injuries
- John treated Mary to a nice dinner
- Mary gave medicine to John for his treatment

Initially, we do the part of speech tagging and Entity named tagging of these sentences $S1$, $S2$ and $S3$. Following Table 6 shows the tagging of all three sentences along with the Table 7 for tags and its description for part of speech tagging and named entity tagging.

We only use the content words (excluding the person's name) for finding semantic relatedness score between two sentences. The content words for first sentence: 'treated', 'injuries'; second sentence: 'treated', 'nice', 'dinner' and third sentence: 'gave', 'medicine', 'treatment'. Here, the proposed method shows how exact sense of the tagged words can be achieved for $S1$ sentence. Table 8 shows different senses present for word 'treated' in WordNet. Table 9 gives the content words present in different senses of 'treated'. For the word 'injuries', senses are present in Table 10 and content words are shown in Table 11. Table 12 and 13 shows different collocation score of each sense of a content word 'treated' and 'injuries' with other content words present in the sentence.

Table 14 gives the correct the senses for all the content words listed below. Correct senses fit with the meaning of the sentence. Here, POS means part of speech.

Table 15 gives different semantic relatedness scores between content words present in the above sentences. Using Algorithm 2, we get the semantic relatedness score among the sentences. Now, to find sense relatedness score, we use Algorithm 3. We find word order information between two sentences using equation 8. All the scores are listed in Table 16. Finally, sense-oriented sentence semantic relatedness score between sentences are calculated using Algorithm 4 for $\alpha = 0,5$, $\beta = 0.3$ and $\gamma = 0.2$ which are empirically determined. Table 17 shows the final scores between sentences. From the above example, it is quite clear that though $S1$ is having more number of common words with $S2$ (Marry; treated; John), but $S1$ is more semantically related with $S3$. Our proposed sense-oriented sentence semantic relatedness measure can distinguish it.

Table 6: Part of speech tagging and named entity tagging list

Word	Tag
Mary	Person
Treated	VBD
John	Person
For	IN
His	PRP$
Injuries	NNS
To	TO
A	DT
Nice	JJ
Dinner	NN
Gave	VBD
Medicine	NN
Treatment	NN

Table 7: Description of tags

Tag	Description
VBD	Verb, past tense
IN	Preposition/sub-conj
PRP$	Personal pronoun
NNS	Noun, plural
TO	"to"
DT	Determiner
JJ	Adjective
NN	Noun, sing. or mass

Table 8: Different senses present for the word 'treated'

Word	Sense number	Meaning
Treat	1	Interact in a certain way
Process	1	Subject to a process or treatment, with the aim of readying for some purpose, improving, or remedying a condition
Treat	3	Provide treatment for
Cover	5	Act on verbally or in some form of artistic expression
Treat	5	Provide with a gift or entertainment
Regale	1	Provide with choice or abundant food or drink
Treat	7	Engage in negotiations in order to reach an agreement
Treat	8	Regard or consider in a specific way

Table 9: Content words present for different senses of the word 'treated'

Word	Sense number	Content
Treat	1	Interact, certain, way
Process	1	Subject, process, treatment, aim, readying, purpose, improving, remedying, condition
Treat	3	Provide, treatment
Cover	5	Act, verbally, form, artistic, expression
Treat	5	Provide, gift, entertainment
Regale	1	Provide, choice, abundant, food, drink
Treat	7	Engage, negotiations, order, reach, agreement
Treat	8	Regard, consider, specific, way

Table 10: Different senses present for the word ‘injuries’

Word	Sense number	Meaning
Injury	1	Any physical damage to the body caused by violence or accident or fracture etc.
Injury	2	An accident that results in physical damage or hurt
Injury	4	An act that causes someone or something to receive physical damage
Injury	5	Wrong doing that violates another’s rights and is unjustly inflicted

Table 11: Content words present for different senses of the word ‘injuries’

Word	Sense number	Content words
Injury	1	Physical, damage, body, caused, violence, accident, fracture
Injury	2	Accident, results, physical, damage, hurt
Injury	4	Act, causes, someone, something, receive, physical, damage
Injury	5	Wrong, doing, violates, another’s, rights, unjustly, inflicted

Table 12: Collocation score of each sense of ‘treated’ with respect to the other words present in the sentences *S*1 and *S*2

Word Pair	Collocation score
(b) Collocation score of the word process with sense number 1	
Injuries-subject	16
Injuries-process	37
Injuries-treatment	320
Injuries-aim	4
Injuries-readying	0
Injuries-purpose	6
Injuries-improving	8
Injuries-remedying	0
Injuries-condition	23
(d) Collocation score of the word cover with sense number 1	
Injuries-act	65
Injuries-verbally	0
Injuries-form	161
Injuries-artistic	0
Injuries-expression	2
(f) Collocation score of the word regale with sense number 1	
Injuries-provide	22
Injuries-choice	18
Injuries-abundant	0
Injuries-food	12
Injuries-drink	0
(g) Collocation score of the word treat with sense number 7	
Injuries-engage	1
Injuries-negotiations	2
Injuries-order	60
Injuries-reach	4
Injuries-agreement	8
(e) Collocation score of the word treat with sense number 5	
Injuries-provide	22
Injuries-gift	0
Injuries-entertainment	3
(g) Collocation score of the word treat with sense number 8	
Injuries-regard	3
Injuries-consider	2
Injuries-specific	31
Injuries-way	48
(c) Collocation score of the word treat with sense number 1	
Injuries-interact	0
Injuries-certain	52
Injuries-way	48
(c) Collocation score of the word treat with sense number 3	
Injuries-provide	22
Injuries-treatment	320

Table 13: Collocation Score of each sense of 'injuries' with respect to the other words present in the sentence S1

Word pair	Collocation score
(a) Collocation score of the word injury with sense number 1	
Treated-physical	47
Treated-damage	16
Treated-body	92
Treated-caused	31
Treated-violence	14
Treated-accident	21
Treated-fracture	17
Treated-etc	25
(b) Collocation score of the word injury with sense number 2	
Treated-accident	21
Treated-results	29
Treated-physical	47
Treated-damage	16
Treated-hurt	3
(c) Collocation score of the word injury with sense number 4	
Treated-act	40
Treated-causes	29
Treated-someone	32
Treated-something	32
Treated-recieve	12
Treated-physical	47
Treated-damage	16
(d) Collocation score of the word injury with sense number 5	
Treated-wrong	17
Treated-doing	5
Treated-violates	0
Treated-rights	33
Treated-unjustly	62
Treated-inflicted	3

Table 14: Accurate senses present for all the content words of *S*1, *S*2 and *S*3 sentences

Word	POS	Sense number	Meaning
Treat	v	03	Provide treatment for
Injury	n	01	Any physical damage to the body caused by violence or accident or fracture etc.
Regale	v	01	Provide with choice or abundant food or drink
Nice	a	01	Pleasant or pleasing or agreeable in nature or appearance
DINNER	n	02	A party of people assembled to have dinner together
give	v	19	Give (as medicine)
Medicine	n	02	(medicine) something that treats or prevents or alleviates the symptoms of disease
Treatment	n	01	Care provided to improve a situation (especially medical procedures or applications that are intended to relieve illness or injury)

Table 15: Semantic relatedness score between two content words present in *S*1 and *S*2, *S*2 and *S*3 and *S*3 and *S*1 sentences

Word pair	Semantic relatedness score
Treat-regale	0
Treat-nice	0
Treat-dinner	0
Injury-regale	0
Injury-nice	0
Injury-dinner	0
Treat-give	6
Treat-medicine	0
Treat-treatment	0
Injury-give	0
Injury-medicine	0
Injury-treatment	0

Table 16: Semantic relatedness, sense relatedness and word order information score between two sentences

Sentence pair	Semantic relatedness score	Sense relatedness score	Word order information score
S1-S2	0	0	0.14
S1-S3	0.375	0.33	0.5
S2-S3	0	0	0.12

Table 17: Sense-oriented sentence semantic relatedness score between two sentences

Sentence pair	Sense-oriented sentence semantic relatedness score
S1-S2	0.028
S1-S3	0.387
S2-S3	0.024

Experimental Analysis and Discussion

Evaluation of Word Sense Disambiguation Technique

We first evaluate our proposed Word Sense Disambiguation (WSD) on publicly available English WSD corpora Sensseval-2, Sensval-3 task1, SemEval-2007 task17, Sem Eval-2013 task 12 and SemEval-2015 task 13 (Raganato *et al.*, 2017). From these two Tables 18 and 19, it signifies that proposed WSD approach works better over many baseline and state-of-the-art systems.

Dataset Description

A standard dataset of 65 noun pairs originally measured by (Rubenstein and Goodenough, 1965) is used widely to evaluate semantic similarity between two words. This dataset has used in many investigations and is established as a stable dataset for finding semantic similarity or relatedness measure (Sánchez *et al.*, 2012). As stated by (Bollegala *et al.*, 2009), to evaluate the accuracy of semantic similarity or relatedness measures is quite tough as it is a subjective human judgment. Rubenstein and Goodenough defined the first experiment as a group of 51 students having English native language. They accessed the similarity of 65 word pairs from ordinary English nouns and scaled it as 0 if pairs have low relatedness and 4 if they have highest relatedness. Miller and Charles (1991) did the same experiment in 1991 with 38 undergraduate students by taking the subset of 30 noun pairs. Resnik (1995) re-created the same experiment. Here, 10 computer-science graduate students and post- graduate students did this experiment. Finally, (Pirró, 2009) replicated the same experiment and compared with three above experiments in 2009 by involving 101 English and non-English native speakers. He got 0.97 average correlation score for Rubenstein and Goodenough experiment and 0.95 for Miller and Charles experiment. This states that similarity value between words are stable and can be used as a reliable source of measures comparison. With the help of this datasets, we find the correlation between human judgment and other similarity measures. Correlation value 1 signifies same similarity with human judgment. This ideal case occurs when one can find similarity perfectly. Whereas, 0 correlation depicts similarity is not related to human judgment. Two correlation coefficients Pearson's and Spearman's are used here. The following two equations are used to find the values:

$$r = \frac{n\left(\sum x_i y_i\right) - \left(\sum x_i\right) - \left(\sum y_i\right)}{\sqrt{\left(n\sum x_i^2\right) - \left(\sum x_i^2\right)}\sqrt{\left(n\sum y_i^2\right) - \left(\sum y_i^2\right)}} \tag{10}$$

The Equation 10 states for Pearson's coefficient; here, x_i corresponds human judgment of *i*-th element and y_i stands for computed value of similarity score for *i*-th element and *n* gives the number of sentence pairs.

Spearman's Coefficient 11 is computed by comparing the sentence similarity measure with human judgment. Here, d_i gives the difference between x_i and y_i:

$$\rho = 1 - \frac{6\sum di^2}{n\left(n^2 - 1\right)} \tag{11}$$

To evaluate our proposed semantic relatedness measure between sentences, we use Pilot Short Text Semantic similarity benchmark dataset by (O'Shea *et al.*, 2008). According to the (Li *et al.*, 2006) explanation, to establish a measure for semantic similarity a survey is conducted by a panel of 32 participants, who are native English speakers. These participants marked the similarity measure for the sentences. Li *et al.* used 30 word couples of (Rubenstein and Goodenough, 1965) and replaced these word couples by their definitions from the Collins Cobuild dictionary (John, 1987).

Another popular dataset MirosoftSoft Parsphrase Corpus (MPSC) is used widely to evaluate whether two sentences are paraphrases or not (Dolan *et al.*, 2004). To evaluate the proposed method, this dataset is used to determine how many paraphrases pairs present in the corpus are correctly identified. At present, MPSC is the largest publicly available paraphrase annotated corpus and is used extensively in evaluation. The matrices used for MPSC corpuse are:

- Recall: It is the ratio of number of determined relevant paraphrases by the proposed method divided by existing number of paraphrases:

$$Recall = \frac{NOP\,corectly\,annotated\,as\,paraphrases}{NOT\ paraphrases\,in\,the\,dataset} \quad (12)$$

Where:
NOP = Number Of Pairs
NOT = Number of Total

- Precision: It is the ratio of number of determined relevant paraphrases by the proposed method divided by number of returned paraphrases by the proposed method:

$$Precision = \frac{NOP\ correctly\ annotated\ as\ paraphrases}{NOP\ annotated\ as\ paraphrases} \quad (13)$$

where, *NOP* is Number Of Pairs

- F-measure: It defines as the mean of recall and precision. F-measure value shows a trade-off between recall and precision measures:

$$F-measure\ =\ 2\times\frac{Precision * Recall}{Precision + Recall} \quad (14)$$

Experiments and Results using Li et al. Dataset

Table 20 gives the mean human sentence similarity values with Proposed semantic relatedness method using Li *et al*. dataset. Our sentence similarity score is compared with mean human similarity. A few portion of the results are presented here. Here, for the 56th *R&G* number, the semantic relatedness value is highest as both sentences contain common words. From the table, it is seen that our proposed sentence relatedness score is much similar with mean human similarity.

Our proposed method's sentence relatedness achieves good Pearson correlation of 0.8987 and Spearman's Coefficient of 0.9091 with mean human similarity. Following Table 21 compares different Pearson's coefficient (r) and Spearman's Coefficient (*r*) values with (Li *et al*., 2006; Islam and Inkpen, 2008; Oliva *et al*., 2011; Taieb *et al*., 2015) algorithms using Li *et al*. dataset. The results got from proposed semantic relatedness measure are competitive and exceeds other existing measures.

Table 18: Performance comparison of different BabelNet-based unsupervised and supervised state-of-the-art methods

Approach	System	SemEval-13	SemEval-15	Macro Avg F1
Unsupervised (Knowledge-based)	Moro 14	66.4	70.3	68.4
	Agirre 14	62.9	63.3	63.1
	Apidianaki 15	-	64.7	-
	Tripodi 17	70.8	-	-
	Wordsim iterSRP2vSim 18	75.0	65.8	70.4
	Proposed WSD	77.8	75.3	76.6
Supervised	Zhong 10	66.3	69.7	68.0
	Weissenborn 15	71.5	75.4	73.5
	Raganato 17	66.9	71.5	69.2
	Pasini 17	65.5	68.6	67.1

Table 19: F-Measure scores of different WSD Methods for all five datasets

Approach	Tr. Corpus	System	Senseval-2	Senseval-3	SemEval-07	SemEval-13	SemEval-15
Supervised	SemCor	IMS	70.9	69.3	61.3	65.3	69.5
		IMS+emb	71.0	69.3	60.9	67.3	71.3
		IMS-S+emb	72.2	70.4	62.6	65.9	71.5
		Context2Vec	71.8	69.1	61.3	65.6	71.9
		MFS	65.6	66.0	54.5	63.8	67.1
	SemCor+OMSTI	IMS	72.8	69.2	60.0	65.0	69.3
		IMS+emb	70.8	68.9	58.5	66.3	69.7
		IMS$_{-s}$+emb	73.3	69.6	61.1	66.7	70.4
		Context2Vec	72.3	68.2	61.5	67.2	71.7
		MFS	66.5	60.4	52.3	62.6	64.2
Unsupervised (Knowledge-based)		Lesk$_{ext}$	50.6	44.5	32.0	53.6	51.0
		Lesk$_{ext}$+emb	63.0	63.7	56.7	66.2	64.6
		UKB	56.0	51.7	39.0	53.6	55.2
		UKB$_{\_gloss}$	60.6	54.1	42.0	59.0	61.2
		Babelfy	67.0	63.5	51.6	66.4	70.3
		WN 1st sense	66.8	66.2	55.2	63.0	67.8
		Proposed WSD	75.4	71.6	63.7	77.8	75.3

Table 20: Sentence relatedness score from proposed method compared with human mean similarity from Li *et al*. dataset

R&G number	Sentence 1	Sentence 2	Mean human similarity	Proposed method sentence relatedness
1	Cord is strong, thick string.	A smile is the expression that you have on your face when you are pleased or amused, or when you are being friendly.	0.01	0.017
3	Noon is 12 o'clock in the middle of the day.	String is thin rope made of twisted threads, used for tying things together or tying up parcels.	0.0125	0.0132
4	Fruit or a fruit is something which grows on a tree or bush and which contains seeds or a stone covered by a substance that you can eat.	A furnace is a container or enclosed space in which a very hot fire is made, for example to melt metal, burn rubbish or produce steam.	0.0475	0.0521
6	An automobile is a car.	In legends and fairy stories, a wizard is a man who has magic powers.	0.0200	0.0345
10	An asylum is a psychiatric hospital	A monk is a member of a male religious community that is usually separated from the outside world.	0.0375	0.0431
16	An asylum is a psychiatric hospital.	A cemetery is a place where dead peoples bodies or their ashes are buried.	0.375	0.245
17	The coast is an area of land that is next to the sea.	A forest is a large area where trees grow close together	0.0475	0.0568
23	A mound of something is a large rounded pile of it.	The shores or shore of a sea, lake, or wide river is the land along the edge of it.	0.0350	0.0321
34	A car is a motor vehicle with room for a small number of passengers.	When you make a journey, you travel from one place to another.	0.0725	0.0845
56	The coast is an area of land that is next to the sea.	The shores or shore of a sea, lake, or wide river is the land along the edge of it.	0.5875	0.5974

Table 21: Experimental comparison of r and ρ values of various algorithms using Li *et al*. dataset

Measure	r	θ
Li *et al*. (2006)	0.81	0.81
STS Islam and Inkpen (2008)	0.85	0.83
SyMSS Oliva *et al*. (2011)	0.76	0.71
FM3S Taieb *et al*. (2015)	0.76	0.79
Proposed method	0.8987	0.9091

By using Li *et al*. dataset, the value of *r* for (Pawar and Mago, 2018) semantic similarity algorithm is 0.8794. It uses 'max similarity' algorithm for word sense disambiguation (Pedersen *et al*., 2005) which is implemented in Pywsd present in NLTK library in Python (Tan, 2014). But this method does not always give the accurate sense based-on the sentence's context. The proposed method also uses more number of relations present in WordNet.

Experiments and Results using MPSC Dataset

Identification of a sentence paraphrase pair depends on the interpretation of proposed sense-oriented sentence semantic relatedness measure ($S1$, $S2$). If semantic relatedness value between (S_1, S_2) >= $\theta \in [0, 1]$, then this sentence pair is considered as a paraphrase. Following Fig. 8 shows quite satisfactory results obtained from the MPSC data. We compare our F-measure value with STS (Islam and Inkpen, 2008)) and FM3S (Taieb *et al*., 2015) measure for different θ values. For each paraphrase pair in the dataset, we consider threshold value θ as 0.6 because we get highest accuracy for $\theta =$ 0.6 as 76.83%. We calculate the semantic relatedness score using Equation 9 and consider it as a paraphrase, if threshold exceeds 0.6.

There is a lack of published research works for providing results on MPSC dataset. Hence, the proposed method is only compared with STS (Islam and Inkpen, 2008) and FM3S (Taieb *et al*., 2015) measures. Following Fig. 6 to 8 represent a comparison of different F-measures among STS, FM3S and proposed approaches according to the different values of θ for the MPSC data. It is quite clear that initially, for smaller values of θ, same F-measure values are for all three measures. As the value of θ increases, proposed method gives best results for $\theta \in [0.4, 1.0]$.

Figures 9 and 10 show the precision and recall values of the proposed sense-oriented sentence semantic relatedness calculation method. In case of precision value, it reaches maximum with $\theta = 0.9$ gives a value of 0.7738. It concludes that pairs of sentence considered as highly related by our proposed sense-oriented sentence semantic relatedness calculation method are treated as paraphrases in MPSC dataset. It is also seen that value of precision is nearly persistent. Actually, proposed method gives high value of relatedness which helps in getting good recall value for threshold $\theta = 1$. In comparison with STS and FM3S methods, for $\theta = 1$; recall value obtained by the proposed method is 0.4883, whereas for STS, it is 0.0054 and for FM3S, it is 0.4557.

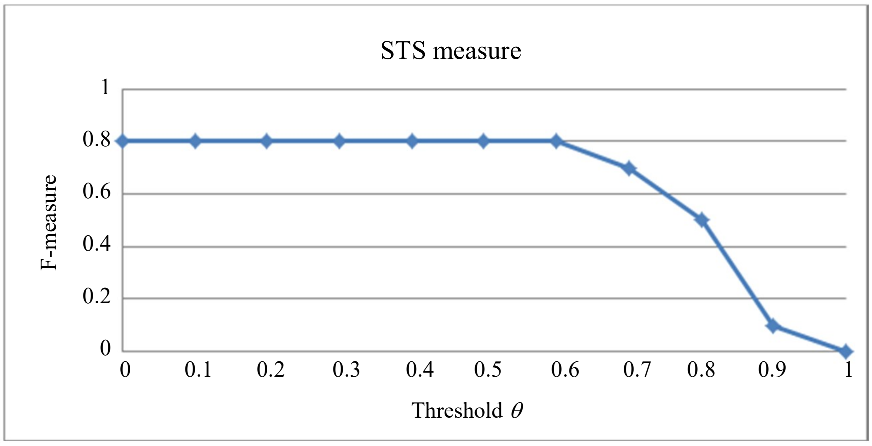

Fig. 6: F-measure curve of STS method applied on MPSC corpus

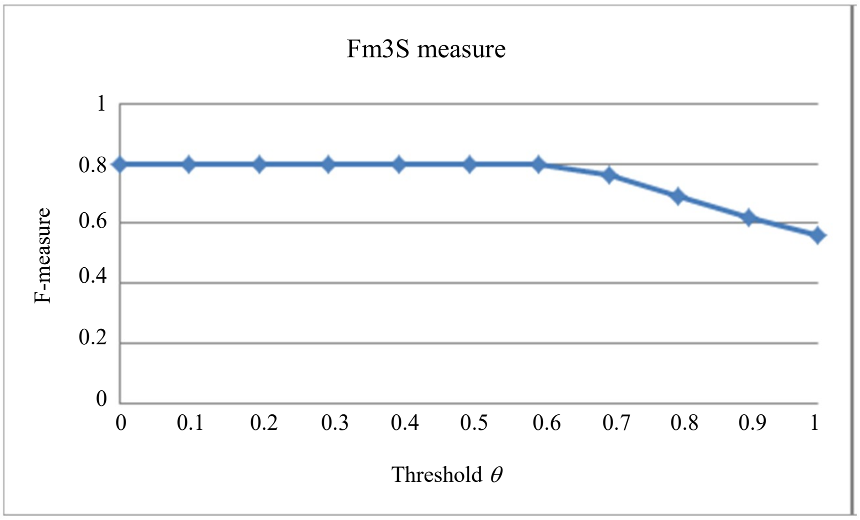

Fig. 7: F-measure curve of FM3S method applied on MPSC corpus

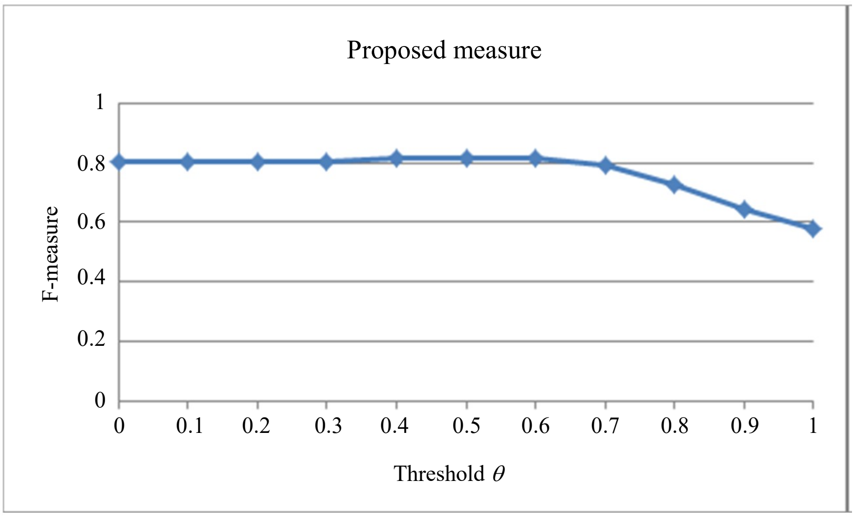

Fig. 8: F-measure curve of Proposed method applied on MPSC corpus

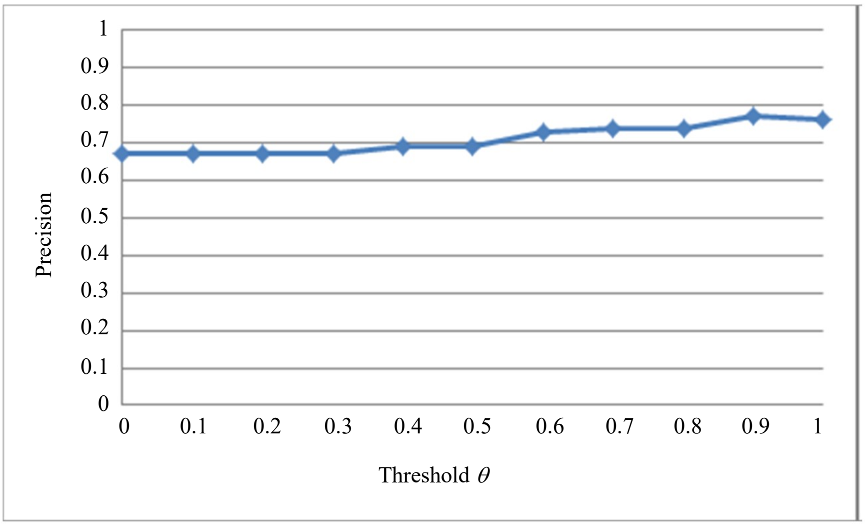

Fig. 9: Precision curve of proposed method applied on MPSC corpus

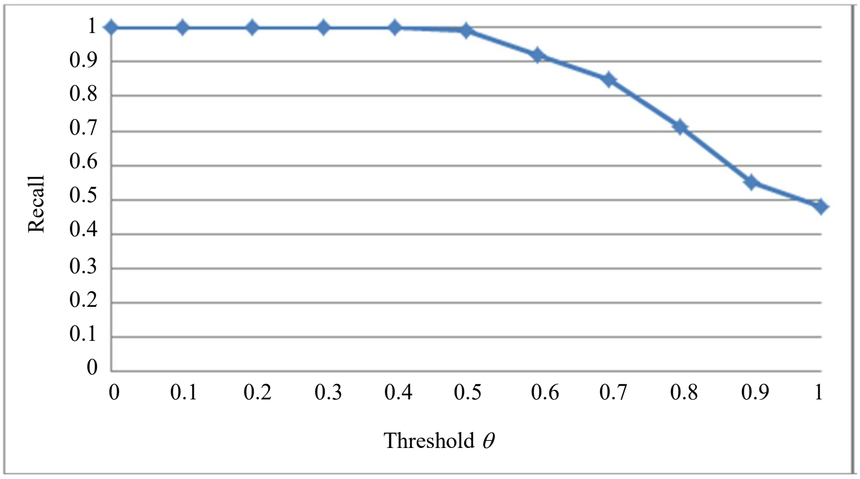

Fig. 10: Recall curve of proposed method applied on MPSC corpus

Table 22: Comparision of different sentence similarity measures using MPSC corpus

Metric	Accuracy	Precision	Recall	F-measure
Prposed method	78.45	75.83	90.0	82.30
PMI-IR (Turney, 2001)	69.9	70.2	95.2	81.0
LSA (Dennis *et al*., 2003)	68.4	69.7	95.2	80.5
STS (Islam and Inkpen, 2008)	72.6	74.7	89.1	81.3
		Semantic Similarity (knowledge-based)		
J & C (Jiang and Conrath, 1997)	69.3	72.2	87.1	79.0
L & C (Leacock and Chodorow, 1998)	69.5	72.4	87.0	79.0
Lesk (1986)	69.3	72.4	86.6	78.9
Li *et al*. (2003)	69.3	71.6	88.7	79.2
W & P (Wu and Palmer, 1994)	69.0	70.2	92.1	80.0
Resnik (1995)	69.0	69.0	96.4	80.4
Combined(S) (Corley and Mihalcea, 2005)	71.5	72.3	92.5	81.2
Combined(U) (Mihalcea *et al*., 2006)	70.3	69.6	97.7	81.3
Baselines Threshold-1 (Mihalcea *et al*., 2006)	33.8	100.0	0.44	0.87
Vector-based (Mihalcea *et al*., 2006)	65.4	71.6	79.5	75.3
Random (Mihalcea *et al*., 2006)	51.3	68.3	50.0	57.8

We also compare our method with other ontology based algorithms for calculating the semantic relatedness score between two text sentences. The Table 22 shows baseline methods and several other methods from (Corley and Mihalcea, 2005) and (Mihalcea *et al*., 2006) on test data. Here, results are also evaluated in terms of accuracy. Accuracy is defined as number of pairs correctly identified as paraphrases by total number of pairs in dataset. We obtain higher F-measure and precision at the cost of decreasing value of recall.

Conclusion and Future Work

Finding accurate semantic relatedness score plays a crucial role which helps in research areas of artificial intelligence. This paper presents a sense-oriented sentence semantic relatedness calculation method for finding the relatedness score between two sentences. First, it finds the sense of each word based-on sentence's context. This appropriate sense helps in finding accurate semantic relatedness score between two words. In fact, while finding relatedness between two sentences, already established relatedness measure does not consider the sense of content words. Hence, we add the particular sense of a word for finding relatedness score. It helps in getting accurate semantic related score. Thus, our semantic relatedness measure does not depend only on lexical knowledge based common human knowledge model, but also able to find sense wise semantic relatedness score. Secondly, we incorporate sense wise relatedness measure by finding number of similar words present in the sentence to the respective sense of each word. It definitely helps in calculating sense-oriented sentence semantic relatedness score between two sentences. Finally, incorporating word order similarity score enhances the accuracy of sense-oriented sentence semantic relatedness score. Different order of words gives different meaning of a sentence. Word order information between two sentences makes an impact to the sequence of same words present in the two sentences. Longest common substring is used here to find word order information between two sentences. Our method can find semantic relatedness score between those sentences in which words are not available in WordNet database. Our method can identify negations.

Our methodology is tested on two widely used benchmark datasets Li *et al.* and MPSC successfully. From the paraphrases detection task on MPSC dataset, proposed method performs well for higher threshold value of $\theta \in [0.6, 1]$. We compare our results with human rated mean similarity values. The results achieved from these experiments conclude that proposed method shows much higher accuracy with improved correlation value of 0.8987 than many existing ontology based methods.

This sentence relatedness measure technique is applicable for short, medium or long type of sentences. This relatedness measure can be used in text clustering, text summarization, plagiarism detection and so on. The key point of this work is the use of sense while finding semantic relatedness between query and input text sentence in query-based text summarization. This sense-oriented sentence semantic relatedness measure helps in extracting more semantically related sentences.

It also helps in solving lexical ambiguity problem. Lexical ambiguity means a word contains more than one meaning. As our proposed semantic relatedness score is based on the sense definition of the each content word present in the sentence, therefore it can also solve the problem of lexical ambiguity. Text summarization and text clustering both are widely used NLP applications. In future, we can use this sense-oriented semantic relatedness measure for creating text summarization and text clustering.

Author's Contributions

Nazreena Rahman: I have proposed the methodology, done the all experiments, coordinated the data-analysis and contributed to the writing of the manuscript.

Bhogeswar Borah: Overall guided how to write the paper and revised the methodology.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Atkinson-Abutridy, J., Mellish, C., & Aitken, S. (2004). Combining information extraction with genetic algorithms for text mining. IEEE Intelligent Systems, 19(3), 22-30. https://doi.org/10.1109/MIS.2004.4

Banerjee, S., & Pedersen, T. (2002, February). An adapted Lesk algorithm for word sense disambiguation using WordNet. In International conference on intelligent text processing and computational linguistics (pp. 136-145). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45715-1_11

Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. www, 7(2007), 757-766. https://dl.acm.org/doi/abs/10.1145/1242572.1242675

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2009, August). A relational model of semantic similarity between words using automatically extracted lexical pattern clusters from the web. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (pp. 803-812). https://doi.org/10.3115/1699571.1699617

Corley, C. D., & Mihalcea, R. (2005, June). Measuring the semantic similarity of texts. In Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment (pp. 13-18). https://doi.org/10.3115/1631862.1631865

Dennis, S., Landauer, T., Kintsch, W., & Quesada, J. (2003). Introduction to latent semantic analysis. In 25th Annual Meeting of the Cognitive Science Society. Boston, Mass (p. 25). http://lsa3.colorado.edu/papers/LSATutorial.pdf

Denoyer, L., & Gallinari, P. (2006, December). The wikipedia xml corpus. In International Workshop of the Initiative for the Evaluation of XML Retrieval (pp. 12-19). Springer, Berlin, Heidelberg. https://doi.org/10.1145/1147197.1147210

Dolan, W., Quirk, C., Brockett, C., & Dolan, B. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. https://doi.org/10.3115/1220355.1220406

Gupta, A., & Yadav, D. K. (2014). Semantic similarity measure using information content approach with depth for similarity calculation. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.643.7991

Hasan, A. M., Noor, N. M., Rassem, T. H., & Hasan, A. M. (2020a). Knowledge-Based Semantic Relatedness measure using Semantic features. International Journal, 9(2). https://doi.org/10.30534/ijatcse/2020/02922020

Hasan, A. M., Noor, N. M., Rassem, T. H., Hasan, A. M., & Hammood, W. A. (2020b, February). A Combined Weighting for the Feature-Based Method on Topological Parameters in Semantic Taxonomy Using Social Media. In IOP Conference Series: Materials Science and Engineering (Vol. 769, No. 1, p. 012002). IOP Publishing. https://doi.org/10.1088/1757-899X/769/1/012002

Hiemstra, D. (2000). A probabilistic justification for using tf×idf term weighting in information retrieval. International Journal on Digital Libraries, 3(2), 131-139. https://doi.org/10.1007/s007999900025

Hirst, G., & St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. WordNet: An electronic lexical database, 305, 305-332.

Islam, A., & Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data (TKDD), 2(2), 1-25. https://doi.org/10.1145/1376815.1376819

Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008. https://arxiv.org/abs/cmp-lg/9709008

John, S. (1987). Looking up: An account of the COBUILD project in lexical computing.

Knappe, R., Bulskov, H., Andreasen, T., & Kaynak, O. (2003). On similarity measures for content-based querying. In 10th International Fuzzy Systems Association World Congress, IFSA (pp. 400-403). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.3708&rep=rep1&type=pdf

Leacock, C., & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. WordNet: An Electronic Lexical Database, 49(2), 265-283.

Lesk, M. (1986, June). 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 (pp. 24-26). https://doi.org/10.1145/318723.318728

Li, Q., Feng, L., Pei, J., Wang, X. S., Zhou, X., & Zhu, Q. (Eds.). (2009). Advances in Data and Web Management: Joint International Conferences, APWeb/WAIM 2009, Suzhou, China, April 2-4, 2009, Proceedings (Vol. 5446). Springer. https://doi.org/10.1007/978-3-642-00672-2

Li, Y., Bandar, Z. A., & McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on Knowledge and Data Engineering, 15(4), 871-882. https://doi.org/10.1109/TKDE.2003.1209005

Li, Y., McLean, D., Bandar, Z. A., O'shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. IEEE Transactions on Knowledge and Data Engineering, 18(8), 1138-1150. https://doi.org/10.1109/TKDE.2006.130

Lord, P. W., Stevens, R. D., Brass, A., & Goble, C. A. (2003). Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. Bioinformatics, 19(10), 1275-1283. https://doi.org/10.1093/bioinformatics/btg153

McKeown, K. R., & Radev, D. R. (2000). Collocations. Handbook of Natural Language Processing.

McRae, K., Khalkhali, S., & Hare, M. (2012). Semantic and associative relations in adolescents and young adults: Examining a tenuous dichotomy. https://psycnet.apa.org/record/2011-12634-002

Meadow, C. T., Boyce, B. R., & Kraft, D. H. (1992). Text information retrieval systems. Academic Press San Diego, CA. https://www.amazon.com/Information-Retrieval-Systems-Charles-Meadow/dp/B002F9OSJS

Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In Aaai (Vol. 6, No. 2006, pp. 775-780). https://www.aaai.org/Papers/AAAI/2006/AAAI06-123.pdf

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. Language and Cognitive Processes, 6(1), 1-28. https://doi.org/10.1080/01690969108406936

Okazaki, N., Matsuo, Y., Matsumura, N., & Ishizuka, M. (2003). Sentence extraction by spreading activation through sentence similarity. IEICE Transactions on Information and Systems, 86(9), 1686-1694. https://search.ieice.org/bin/summary.php?id=e86-d_9_1686

Oliva, J., Serrano, J. I., Del Castillo, M. D., & Iglesias, Á. (2011). SyMSS: A syntax-based measure for short-text semantic similarity. Data & Knowledge Engineering, 70(4), 390-405. https://doi.org/10.1016/j.datak.2011.01.002

O'Shea, J., Bandar, Z., Crockett, K., & McLean, D. (2008). Pilot short text semantic similarity benchmark data set: Full listing and description. Computing.

Park, J., Kim, K., Hwang, W., & Lee, D. (2019). Concept embedding to measure semantic relatedness for biomedical information ontologies. Journal of Biomedical Informatics, 94, 103182. https://doi.org/10.1016/j.jbi.2019.103182

Patwardhan, S., Banerjee, S., & Pedersen, T. (2003, February). Using measures of semantic relatedness for word sense disambiguation. In International conference on intelligent text processing and computational linguistics (pp. 241-257). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36456-0_24

Pawar, A., & Mago, V. (2018). Calculating the similarity between words and sentences using a lexical database and corpus statistics. arXiv preprint arXiv:1802.05667. https://arxiv.org/abs/1802.05667

Pedersen, T., Banerjee, S., & Patwardhan, S. (2005). Maximizing semantic relatedness to perform word sense disambiguation. Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute.

Pedersen, T., Pakhomov, S. V., Patwardhan, S., & Chute, C. G. (2007). Measures of semantic similarity and relatedness in the biomedical domain. Journal of Biomedical Informatics, 40(3), 288-299. https://doi.org/10.1016/j.jbi.2006.06.004

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004, July). WordNet:: Similarity-Measuring the Relatedness of Concepts. In AAAI (Vol. 4, pp. 25-29). https://doi.org/10.3115/1614025.1614037

Perkins, J. (2014). Python 3 text processing with NLTK 3 cookbook. Packt Publishing Ltd.

Pesquita, C., Faria, D., Falcao, A. O., Lord, P., & Couto, F. M. (2009). Semantic similarity in biomedical ontologies. PLoS comput biol, 5(7), e1000443. https://doi.org/10.1371/journal.pcbi.1000443

Petrakis, E. G., Varelas, G., Hliaoutakis, A., & Raftopoulou, P. (2006). X-similarity: Computing semantic similarity between concepts from different ontologies. Journal of Digital Information Management, 4(4). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.464.6308&rep=rep1&type=pdf

Pirró, G. (2009). A semantic similarity metric combining features and intrinsic information content. Data & Knowledge Engineering, 68(11), 1289-1308. https://doi.org/10.1016/j.datak.2009.06.008

Plaut, D. C. (1995). Semantic and Associative Primi Distributed Attractor Network. In Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society (Vol. 17, p. 37). Psychology Press.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics, 19(1), 17-30. https://doi.org/10.1109/21.24528

Raganato, A., Bovi, C. D., & Navigli, R. (2017, September). Neural sequence learning models for word sense disambiguation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 1156-1167). https://doi.org/10.18653/v1/D17-1120

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007. https://arxiv.org/abs/cmp-lg/9511007

Richardson, R., Smeaton, A., & Murphy, J. (1994). Using WordNet as a knowledge base for measuring semantic similarity between words. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.49.6027&rep=rep1&type=pdf

Rodriguez, M. A., & Egenhofer, M. J. (2003). Determining semantic similarity among entity classes from different ontologies. IEEE transactions on knowledge and data engineering, 15(2), 442-456. https://doi.org/10.1109/TKDE.2003.1185844

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. Communications of the ACM, 8(10), 627-633. https://doi.org/10.1145/365628.365657

Sadr, H., Soleimandarabi, M. N., Pedram, M., & Teshnelab, M. (2019, April). Unified topic-based semantic models: A study in computing the semantic relatedness of geographic terms. In 2019 5th International Conference on Web Research (ICWR) (pp. 134-140). IEEE. https://doi.org/10.1109/ICWR.2019.8765257

Sánchez, D., Batet, M., Isern, D., & Valls, A. (2012). Ontology-based semantic similarity: A new feature-based approach. Expert systems with applications, 39(9), 7718-7728. https://doi.org/10.1016/j.eswa.2012.01.082

Slimani, T. (2013). Description and evaluation of semantic similarity measures approaches. arXiv preprint arXiv:1310.8059. https://arxiv.org/abs/1310.8059

Taieb, M. A. H., Aouicha, M. B., & Bourouis, Y. (2015, June). Fm3s: Features-based measure of sentences semantic similarity. In International Conference on Hybrid Artificial Intelligence Systems (pp. 515-529). Springer, Cham. https://doi.org/10.1007/978-3-319-19644-2_43

Tan, L. (2014). Pywsd: Python Implementations of Word Sense Disambiguation (WSD) Technologies.

Turney, P. D. (2001, September). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In European conference on machine learning (pp. 491-502). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44795-4_42

Tversky, A. (1977). Features of similarity. Psychological review, 84(4), 327. https://doi.org/10.1037/0033-295X.84.4.327

Wiemer-Hastings, P. (2000). Adding syntactic information to LSA. In Proceedings of the Annual Meeting of the Cognitive Science Society (Vol. 22, No. 22). https://escholarship.org/content/qt7r89x190/qt7r89x190.pdf

Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. arXiv preprint cmp-lg/9406033. https://doi.org/10.3115/981732.981751

Zhou, Z., Wang, Y., & Gu, J. (2008, November). New model of semantic similarity measuring in wordnet. In 2008 3rd International Conference on Intelligent System and Knowledge Engineering (Vol. 1, pp. 256-261). IEEE. https://doi.org/10.1109/ISKE.2008.4730937

Zhu, X., Guo, Q., Zhang, B., & Li, F. (2019). An efficient approach for measuring semantic relatedness using Wikipedia bidirectional links. Applied Intelligence, 49(10), 3708-3730. https://doi.org/10.1007/s10489-019-01452-1