Journal of King Saud University – Computer and Information Sciences 34 (2022) 6643–6651

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

An unsupervised method for word sense disambiguation

Nazreena Rahman [a,*], Bhogeswar Borah [b]

[a] Department of Computer Science and Engineering, Assam Kaziranga University, Jorhat, Assam 785006, India
[b] Department of Computer Science and Engineering, Tezpur University, Sonitpur, Assam 784028, India

ARTICLE INFO

Article history:
Received 26 February 2021
Revised 6 July 2021
Accepted 29 July 2021
Available online 6 August 2021

Keywords:
Word sense disambiguation (WSD)
WordNet lexical dictionary
Wikipedia corpus
Collocation extraction score
Knowledge-based unsupervised and supervised systems
Senseval and SemEval datasets

ABSTRACT

Word sense disambiguation (WSD) finds the actual meaning of a word according to its context. This paper presents a novel WSD method 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. Initially, we find all the probable senses of the target word using WordNet. For each of the words present in a sense, we calculate the collocation extraction score with the other words in the sentence. The collocation extraction score finds the probability of the occurrence of two words together in the Wikipedia corpus. The maximum collocation extraction score assigns the proper sense for that context of the sentence. Our method is not limited to the bi-grams that are made up of only two consecutive words. Our method can find the probability of having two words together in a sentence when other words separate these two words. To compare our WSD method with current knowledge-based unsupervised and supervised systems, we use different Senseval and SemEval datasets for doing WSD on English words. Finally, the experimental analysis illustrates the significance of the proposed approach over many baseline and current systems.

1. Introduction

Removal of ambiguity from words is an emerging problem in natural language processing and ontology. In computational linguistics, most languages are polysemous. The word *bank* is said as an ambiguous word as it has different meanings for the context of the sentence. For example, in "I am sitting on the bank of the river", the bank says about sloping land. In "I went to the bank to transfer money", the bank stands for a financial institution. We can predict the sense of a word by immediately referring to the other content words present in the sentence. In the case of computer programs, it is more challenging to find out the exact sense of a word. The process of finding an accurate sense of a polysemous word is known as WSD. WSD is a significant component applied in many fields like text analytics, information retrieval, text mining, speech recognition.

WSD technique is applicable in solving many natural language processing tasks like text summarization, question–answering, information retrieval, and text classification. For finding semantic relatedness between two sentences, the proposed technique can be applied to get exact semantic similarity or relatedness score. Semantic similarity or relatedness score is computed using WordNet.

1.1. Introduction to Wordnet

Unlike the traditional dictionary, WordNet Miller, 1995 is a lexical database where content words are organized semantically. Content words include nouns, verbs, adverbs and adjectives. WordNet relations are applied widely in text analysis and artificial intelligence applications. This lexical database is used for English language and was created by Cognitive Science Laboratory of Princeton University. This richer structure 'WordNet' has a set of synonymous words or same lexical category words, known as *synset* or *synonym set*. They carry the same meaning words. For example *motorcar* has a synset *car*. In WordNet, concepts are represented by the word, its part of speech and its sense number. For example: for the word *bank*, the concept is *bank#n#1*. It means the word *bank* is a noun here and it has the first sense. Concepts in WordNet are linked together in a hierarchical structure. Different kind of synsets are related by different semantic relations in WordNet. In English language, most of the words have more than one

* Corresponding author.
E-mail addresses: nazreena.rehman@gmail.com, nilufarnew@gmail.com (N. Rahman).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.07.022
1319-1578/

senses. If a word has more than one synsets, then that word is a polysemous word. For example *bank* sometimes used to represent a financial institution or can be used for slopping land. Each content word in a synset has its gloss or definition. In fact, in WordNet, maximum content words have more than one sense definition.

In general, with the help of the WordNet database, we find the semantic similarity or relatedness between two content words. Here, similarity and relatedness both are distinctly apart. Two words are said as similar words if they are synonyms or can substitute the words with each other in the same context. The relatedness measure between two words gives a larger set of a relationship (synonym, hypernym, troponym, antonym etc.) between the two words. For example, car and gasoline are more semantically related whereas car and automobile both are similar. Word relatedness contains much broader relations than similarity (Martin and Jurafsky, 2009). Different types of relations are present in the WordNet database. Table 1 shows different semantic relations with its meaning.

1.2. Introduction to Wikipedia Corpus

Moreover, in recent times, the corpus-based technique to disambiguate a word's sense is quite popular. It finds the similarity between words using the information exclusively derived from large corpora. A combination of both knowledge-based and corpus-based WSD methods help in getting better performance in WSD. We have proposed one WSD method in which Wikipedia corpus is used to disambiguate a word. Natural language processing tasks require is a corpus. This Wikipedia corpus contains the full text data of Wikipedia. It consists of 1.9 billion words in more than 4.4 million articles. This corpus allows us to search Wikipedia in a more powerful way. We are taking this dataset for finding collocation score as Wikipedia is a rich source of well-organized textual data. It is also a vast collection of knowledge. We have build a corpus from the set of English articles. It is freely available. We have used Gensim libary for building the corpus. Initially, we find all possible senses using WordNet and then we calculate the probability of occurrence of two words together in the Wikipedia corpus. We use collocation-based feature to find the same sense of words. Collocation means the probability of occurrence or use of two words together in a corpus. These are the words that often go together. This exact sense of words of query and input text sentence will be further used to find semantic relatedness score. Exact word sense will help in extracting meaningful query-related sentences for query-based text summarization.

1.3. Contributions

Our main contributions are as follows: (1) We provide a word sense disambiguation method to find the correct sense of a word. This method is not limited to the bi-grams that made up of only two consecutive words. Senses of the words help in getting accurate meaning of the complete sentence. (2) The WSD method proposed helps get accurate relatedness measures between two words and between two sentences. To our best knowledge, inserting the correct sense is not incorporated directly while finding semantic relatedness score.

Table 1
Semantic Relations among Senses in WordNet.

Relation Name	Meaning	Example
Synonymy:	identical or nearly identical	car/ automobile
Hyponymy:	one sense is subclass of other	car and vehicle
Hypernym:	one sense is the superordinate class of another	vehicle and car
Meronymy:	the part-whole relation	the leg is a part of the chair
Holonymy:	whole to part relation	body and hand
Troponymy:	manner of doing something	stroll and walk
Entailment:	deduction or implication	tease and disappoint
Antonymy:	senses with the opposite meaning	up/down

2. Existing works on disambiguation of words

In recent times, corpus based techniques to disambiguate the word's sense is quite popular. Human has the experience of a text corpus. In this approach, we need to label each word with a tag. This manually created sense-tagged text is used for word sense disambiguation. This tag gives the definition of that word which is most proper for the sentence. This is a supervised approach. This sense-tagged text works as a training set which can be used to disambiguate the unknown word's sense. The main drawback of this approach is to create the sense-tagged text (Banerjee and Pedersen, 2002).

Different well known and popular supervised and knowledge-based techniques are implemented to disambiguate a word sense. Here, we have mentioned a few popular supervised WSD techniques: Decision tree algorithm is widely used by many researchers (Singh et al., 2014) in which decision tree is used to denote classification rules in a tree structure where training dataset is recursively divided. Each leaf node denotes the sense of a word. Pederson (Pedersen, 2001) uses a corpus based approach where a sense to a ambiguous word is assigned by a decision tree based on the bigram that occurs nearby. Another supervised approach is proposed by O'Hara et al. (O'Hara et al., 2004) where class based collocation is used for word sense disambiguation. Three different word relatedness scores are used for the collocation: first one is WordNet hypernym relations; second one is cluster-based word similarity classes; and third one is dictionary definition analysis. Another supervised method is proposed by Popov (Popov, 2017) where recurrent neural network is used. This model is based on LSTM cells. LSTM stands for Long Short Term Memory. This LSTM helps to capture word order information and adds distributed word representations (embeddings) as features. Another LSTM based semi-supervised word sense disambiguation method is proposed by Yuan et al. (2016). Further, results of the reproduction study of this method is done by Le et al. (2018). They have done a deeper examination of the effect of various factors on its performance. A number of interesting findings are derived. First is that a very large unannotated dataset is not needed to get state-of-the-art WSD performance as they have used Gigaword corpus. It is also observed that this approach has a more balanced sense assignment in comparison with other techniques. It is shown by the relatively good performance on less-frequent-sense instances. Additionally, it is also identified that limited sense coverage in annotated dataset places a potentially upper bound for the overall performance. Support Vector Machine (SVM) (Zhong and Ng, 2010) classifier is used in It Makes Sense (IMS) Word Sense Disambiguation (WSD) system. Different features like: surrounding words, PoS tag of surrounding words and local collocations are taken. In the works of (Taghipour and Ng, 2015; Rothe and Schütze, 2015; Iacobacci et al., 2016), word embeddings is used. Different methods are proposed by (Iacobacci et al., 2016) where word embedding is used in current supervised WSD systems. They have made a deep analysis how different parameters are affecting the performance of WSD system. Here, two best configurations are considered having one with surrounding words (IMS + emb) and other one without surrounding words (IMS_-s+emb). In both methods, they integrates word embeddings by using exponential decay. To train the word embeddings, Iacobacci et al.'s suggested learning strategy and hyper parameter's are used (Raganato et al., 2017). Nowadays, Neural language based model is used widely for WSD task (Melamud et al., 2016; Kågebäck and Salomonsson, 2016; Yuan et al., 2016). Experiment uses bidirectional LSTM (Melamud et al., 2016) model. This context2vec neural model learns a generic embedding function for variable length contexts of target words.

Knowledge-based system includes following three Word Sense Disambiguation models: Lesk is simple knowledge based WSD algorihm (Lesk, 1986) which finds similar words between the definition of each sense with context of the target word. For comparison purpose, extended version of Lesk algorithm is used where definition of related senses are also included. Here, for word weighting, conventional term frequency-inverse document frequency is used (Jones, 1972; Banerjee and Pedersen, 2003). For better analysis, word embedding is added in enhanced version of Lesk which helps in computing similarity between definition and context of the target word (Basile et al., 2014). A graph based WSD system is proposed by Agirre et al. (Agirre and Soroa, 2009) where random walk is used over a WordNet semantic network (Agirre et al., 2014). In their method, a personalized Page Rank algorithm (Haveliwala, 2002) is used. Babelfy is a graph-based WSD approach where random walk is used to find connections between synsets (Moro et al., 2014). Babelfy uses random walks with restart (Tong et al., 2006) over BabelNet (Navigli and Ponzetto, 2012). Babelfy includes the whole document while finding its sense.

From the above literature survey, it is found that many supervised, unsupervised and knowledge-based approaches are available to disambiguate an ambiguous word. A word has multiple senses at different context of the sentence, our aim is to disambiguate the meaning of that target word for that context. Our proposed method is different from the existing graph based methods (Navigli and Lapata, 2009; Corrêa and Amancio, 2019; Arab et al., 2016) particularly in that case that we are using collocation score to give scores to different senses present for a word. Collocation score is calculated with the help of Wikipedia corpus. Initially, we find the all possible senses of a word using WordNet Dictionary. Proposed method uses bi-gram collocation by using Wikipedia corpus. The main contribution of this chapter is that proposed method is applicable for finding semantic relatedness score between two sentences as it disambiguates a word for the sentence.

3. 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. Gloss implies a dictionary-style 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.

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, 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 automatically takes that sense for which it gets the highest semantic similarity score. Using WordNet 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. Tables 3 provides the different senses present for the word *bank*. For the word *river* only one noun sense is present in the WordNet.

Table 4 shows the trace definition present for *bank#n#1*, *bank#n#2*, *bank#n#3* and *river#n#1*. Trace definition shows how

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 et al., 1998	0
The relatedness of bank#n#1 and river#n#1 using Adapted lesk Banerjee and Pedersen, 2002	16
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
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

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.

4. Proposed Unsupervised Method for Word Sense Disambiguation

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

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. 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 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.
  We use NLTK tool for pre-processing purpose as NLTK is great for pre-processing and tokenizing text. NLTK is widely used as a teaching and research tool. It supports teaching and research in NLP. It is a platform for prototyping and building different research systems. Gensim is mainly used for topic modeling and document similarity. Standford CoreNLP is written in Java, but as I am using Python language, therefore I will prefer NLTK. Stadford CoreNLP is only for tokenizing and POS tagging and it also requires more resources. SpaCy is a huge memory hog. In case of sentence tokenization, NLTK outperforms spaCy. However, spaCy is not also a research software.
- 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.

4.1. 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 same 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.

WordNet is a part of NLTK for Python. This extensive library makes the natural language processing easy.The proposed 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 method finds all the senses of the target words present in the WordNet. We use WordNet dictionary as it gives all the possible senses which should be present in a word. In future we will use this sense to calculate the semantic relatedness score. 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 Eq. 1. Here, Wikipedia corpus is used (Denoyer and Gallinari, 2006).

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

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 method. We will search for bi-gram frequency where the words may be separated by three other words in the text of Wikipedia. The corpus contains full text data of Wikipedia. Therefore, possibility of absence of words targeted in the corpus or is almost zero. However, if the words targeted are not together with the span size used (i.e., 3), the method will change the span size to 9. We are using the huge Wikipedia corpus. Hence, there is no such possibility for getting the collocation score NULL. 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:

$$CS(Sense, Sentence) = max \sum_{w \in Sense, w\prime \in Sentence} (collocation\,score(w, w\prime)) \tag{2}$$

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.

4.2. Finding Exact Sense of a Word Present in a Sentence

The proposed 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:

Table 5
Highest collocation score for the content words present in each verb sense of 'treated' with respect to the other content words present in the sentence *S*1.

Word	Sense Number	Word Pair	Collocation Score
treat	1	certain-injuries	52
process	1	treatment-injuries	320
treat	3	treatment-injuries	320
cover	5	form-injuries	161
treat	5	provide-injuries	22
regale	1	provide-injuries	22
treat	7	order-injuries	60
treat	8	way-injuries	48

sense of a word with respect to the context of the sentence. The content words for first sentence: 'treated','injuries'; second sentence: 'treated','dinner' and third sentence: 'prime','minister', 'visit', 'Assam'. For each content word, we get different senses. For each sense, we find the content words.

Table 5 shows highest collocation scores of each sense of *'treated'*. For example: the meaning of the first sense of *'treated'* is

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 3.2
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
```

51

5. Implementation of the proposed word sense disambiguation method

To illustrate the implementation of the proposed method on different sentences, we have given below a detailed description of the proposed method with three different ambiguous *S*1, *S*2 and *S*3 sentences:Mary treated John for his injuries, John treated Mary to dinner and Prime Minister's visit to Assam.

Initially, we do the part of speech tagging (POS) and named entity tagging (NET) of these sentences *S*1, *S*2 and *S*3. We only use the content words (excluding the person's name) for finding *'interact in a certain way'*. We have found the content words present in the meaning. They are: *'interact', 'certain' and 'way'*. When we have calculated the collocation score between *'interact', 'certain' and 'way'* with *'injuries'*, we got highest collocation score 52 for *'injuries'* with *'certain'*. It is quite clear that for the word *'treated'*, from all the senses we get highest collocation score 320 for the sense: process#v#1 and treat#v#3. We take the process#v#1 as it has the lower sense number. Similarly, for the word injuries, from the all senses we get highest collocation score 92 for the sense injury#n#1 which fits with the meaning of the sentence.

In case of second sentence, for the word *'treated'*, we get the highest collocation score 100 for regale#v#1 sense. We can say that though *'treated'* word is used in both *S*1 and *S*2 sentences, but they carry different meaning for both sentences. Therefore, our proposed method can distinguish it clearly. Similarly, highest collocation score for *'dinner'* is 723 for the sense number 2.

Finally, for third sentence *S*3, highest collocation scores for each sense of *'prime'*, *'minister'* and *'visit'* are 2227, 11094 and 6610. Therefore, from the all senses of *'prime'*, we get the highest collocation score for the sense prime#n#4. Similarly, for *minister* and *visit*, appropriate senses are: minister#n#4 and visit#n#1.

6. Experimental analysis and discussion

6.1. Dataset

Our proposed method is evaluated on publicly available English WSD corpora Senseval-2, Senseval-3 task 1, SemEval-2007 task 17, SemEval-2013 task 12 and SemEval-2015 task 13 (Raganato et al., 2017). Two sense-annotated WordNet corpuses SemCor (Miller et al., 1994) and OMSTI (Taghipour and Ng, 2015) are used for training the supervised system for our evaluation and comparison purpose.

6.2. Comparison systems

Different current and widely used supervised and knowledge-based word sense disambiguation systems are included for comparison purpose. All the supervised systems use the same corpus SemCor and Semcor + OMSTI for training purpose. This gives a fair comparison. For the supervised system, a baseline method is included in Table 6 by taking the Most Frequent Sense (MFS) heuristically. Senses are selected based on the highest number of occurrences in the training corpus (Raganato et al., 2017). Similarly, in case of Knowledge-based system, a baseline method is also chosen to select first sense as a correct sense present in WordNet 3.0 (Bird et al., 2009). All the existing semantic similarity or relatedness measure uses the first sense while finding its semantic similarity or relatedness score. We use F-Score criteria for evaluation purpose. F-Score is the harmonic mean of precision and recall value.

6.3. Comparison with different supervised and knowledge-based systems

Table 6 compares F-Measure values of different WSD systems as mentioned above. Two different category of have been used here: one is supervised another one is knowledge-based unsupervised. For the supervised approach, different machine learning models are trained by a corpus (Dongsuk et al., 2018) in which human annotator annotate the correct senses of ambiguous words (Weissenborn et al., 2015; Melamud et al., 2016; Raganato et al., 2017). From the survey, it is found that construction of training corpus for all the languages and word is quite an expensive work. Hence, supervised approaches have some limitations on the set of disambiguated words. On the contrary, knowledge-based WSD system uses lexical dictionary like WordNet (Banerjee and Pedersen, 2003; Chaplot et al., 2015). Both contextual information and semantic knowledge have been incorporated in knowledge-based WSD systems. Hence, more number of words can be disambiguated using knowledge-based approach. Conclusion can be derived that knowledge-based WSD systems are more practically feasible and suitable than supervised WSD systems (Chaplot et al., 2015; Dongsuk et al., 2018).

Results in the Table 6 clearly shows that though many supervised methods outperforms existing knowledge-based methods, but our proposed method has performed much better for different datasets. Existing knowledge-based methods for word sense disambiguation are not up-to the mark though knowledge-based systems are more practically feasible and suitable. Their methods have limitations that is why we have put forward an improved word sense disambiguation method where both WordNet dictionary and Wikipedia corpus are used. It is also noticed from the Table that the performance of WSD methods for all datasets are not uniform. A large performance gap is seen between the best and worst performing dataset. For dataset SemEval-07, performance is quite low for all WSD systems as this dataset is the most ambiguous one.

We compare our proposed approach to other BabelNet-based unsupervised and supervised WSD systems (Dongsuk et al., 2018). BabelNet is a multilingual lexical semantic network. It is automatically created by linking Wikipedia to the WordNet (Navigli and Ponzetto, 2012). From experimental results, it is seen that our proposed method outperforms all listed WSD systems for SemEval-2013 dataset. For the SemEval-2015 dataset, our proposed method has similar performance to the supervised Weissenborn et al. method. However, in terms of macro average score of both datasets SemEval-13 and SemEval-15, the proposed method shows higher performance for all WSD systems present in the Table 7. A macro-average calculates the metric independently for each class and then takes the average value.

We compare our WSD Method with other recognized and current state-of-the-art existing word sense disambiguation methods where SENSEVAL-2 dataset is used (Wiriyathammabhum et al.,

Table 6
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 (Zhong and Ng, 2010)	70.9	69.3	61.3	65.3	69.5
		IMS + emb (Iacobacci et al., 2016)	71.0	69.3	60.9	67.3	71.3
		IMS-S + emb (Raganato et al., 2017)	72.2	70.4	62.6	65.9	71.5
		Context2Vec (Melamud et al., 2016)	71.8	69.1	61.3	65.6	71.9
		MFS (Raganato et al., 2017)	65.6	66.0	54.5	63.8	67.1
	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
		$Lesk_{ext}$ (Banerjee and Pedersen, 2003)	50.6	44.5	32.0	53.6	51.0
		$Lesk_{ext}$+emb (Basile et al., 2014)	63.0	63.7	56.7	66.2	64.6
		UKB (Agirre et al., 2014)	56.0	51.7	39.0	53.6	55.2
Unsupervised		$UKB_{\_gloss}$ (Agirre et al., 2014)	60.6	54.1	42.0	59.0	61.2
(Knowledge-based)		Babelfy (Moro et al., 2014)	67.0	63.5	51.6	66.4	70.3
		WN 1^{st} sense (Bird et al., 2009)	66.8	66.2	55.2	63.0	67.8
		Proposed Method	**75.4**	**71.6**	**63.7**	**77.8**	**75.3**

Table 7
Performance comparison of different BabelNet-based unsupervised and supervised state-of-the-art methods

Approach	System	F-score for SemEval-13	F-score for SemEval-15	Macro Avg F-score
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_iter$_{SRP2vSim}$ 18	75.0	65.8	70.4
	Proposed Method	**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 8
Recall values of various WSD Methods.

Method Name	Recall-Score
MFS	47.60%
1-NN	43.11%
PCA	44.45%
KPCA (polynomial)	37.50%
KPCA (Gaussian RBF)	47.71%
NB	49.95%
Logistic Regression	60.07%
MLP	59.70%
Linear SVM	60.40%
SVM (polynomial)	47.71%
SVM (Gaussian RBF)	51.02%
DBN	61.30%
Proposed Method	**76.80%**

2012). In the works of Wiriyathammabhum et al. (Wiriyathammabhum et al., 2012), they have used three different features: topical feature; local feature and part of speech-feature. They have implemented SENSEVAL-2 dataset on the following learning methods to disambiguate a word: (1) Naive Bayes, (2) Nearest Neighbor, (3) Principal Component Analysis, (4) Kernel Principal Component Analysis, (4) Logistic Regression, (5) Multi-layer Perceptron (6) Support Vector Machine and (7) Deep Belief Networks. They have chosen MFS (Most Frequent Sense) as baseline method. From the Table 8 it is seen that proposed WSD Method shows considerably better results. The proposed method also improves the task in a theoretical way as because for any Word Sense Disambiguation dataset, our method performs a better result. From the Table 6, it shows that the average F-Measure for all the Senseval and SemEval datasets is 72.76% which is the highest average value than all the considered knowledge-based and supervised methods. Similarly, for BabelNet based supervised and unsupervised methods, it is seen from the Table 7 that the average F-score is 76.6 %, which is also highest average value amongst all listed methods.

7. Statistical significance test

To statistically compare the performance of our proposed sentence similarity measure with other Word Sense Disambiguation methods, we use a non-parametric Wilcoxon's matched-pairs signed rank based statistical significance test. It helps in determining the significance of our results. The statistical significance test for independent samples has been conducted at the 5% significance level of the results. Two groups are compared here; one corresponding to our proposed method and the other corresponding to existing considered method. Each method consists of F-scores for the SemEval-13 dataset. To establish that this goodness is statistically significant, we have found the P values produced by

Table 9
P values produced by Wilcoxon's matched-pairs signed rank test by comparing Proposed Method with other methods for SemEval-13 dataset.

Approach	System	P-value
Supervised	IMS (Zhong and Ng, 2010)	0.036
	IMS + emb (Iacobacci et al., 2016)	0.040
	IMS-S + emb (Raganato et al., 2017)	0.037
	Context2Vec (Melamud et al., 2016)	0.037
	MFS (Raganato et al., 2017)	0.034
	0.036	
	IMS + emb	0.038
	IMS$_{_s}$+emb	0.039
	Context2Vec	0.040
	MFS	0.033
	Zhong 10 (Zhong and Ng, 2010)	0.038
	Weissenborn 15 (Weissenborn et al., 2015)	0.039
	Raganato 17 (Raganato et al., 2017)	0.039
	Pasini 17 (Pasini and Navigli, 2017)	0.037
	Lesk$_{ext}$ (Banerjee and Pedersen, 2003)	0.000
	Lesk$_{ext}$+emb (Basile et al., 2014)	0.038
	UKB (Agirre et al., 2014)	0.000
Unsupervised	UKB$_{_gloss}$ (Agirre et al., 2014)	0.001
	WN 1st sense (Bird et al., 2009)	0.034
(Knowledge-based)	Babelfy (Moro et al., 2014)	0.038
	Moro 14 (Moro et al., 2014)	0.038
	Agirre 14 (Agirre et al., 2014)	0.033
	Tripodi 17 (Tripodi and Pelillo, 2017)	0.042
	Wordsim_iter$_{SRP2vSim}$ 18 (Dongsuk et al., 2018)	0.042

Wilcoxon's matched-pairs signed rank test for comparison of two groups. As a null hypothesis, it is assumed that there are no significant differences between the median values of two groups. Whereas the alternative hypothesis is that there is significant difference in the median values of the two groups. Table 9 shows that the Wilcoxon's matched-pairs signed rank test between our proposed method and existing method for SemEval-13 corpus provides a 5% less P value, which is very small. This is strong evidence against the null hypothesis, indicating that the better median values of the performance metrics produced by our proposed method is statistically significant and has not occurred by chance. It establishes the significant superiority of the proposed method. It is clear that P value is much less than 0.05 (5% significance level). From the statistical results, we observe that our proposed method significantly outperforms existing methods. Additionally, we can say that our proposed method is more stable than the other methods.

8. Discussion on performance of query relevance

Our proposed method performs better as we are using collocation score to disambiguate a word. The proposed method can find

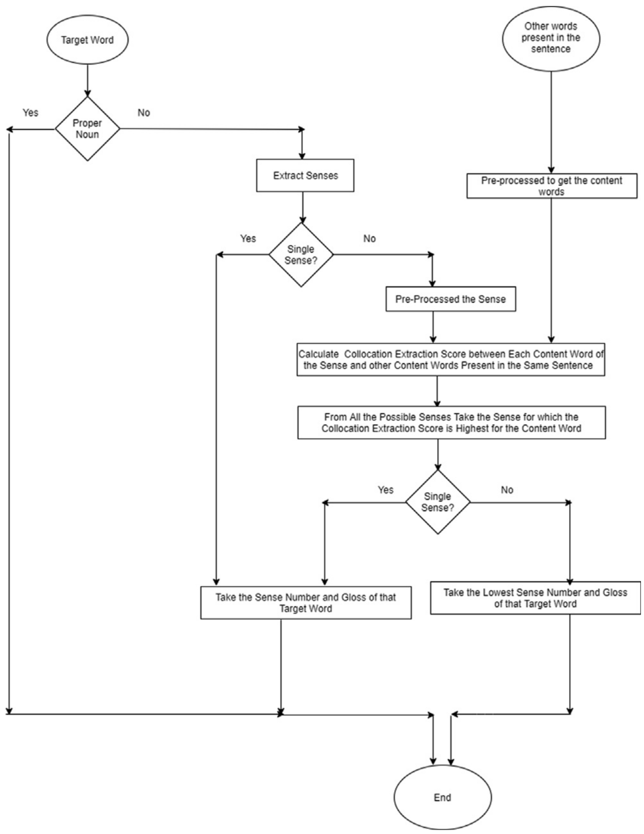

Fig. 1. Block Diagram of Word Sense Detection Method.

out the collocation score between multi-words terms. We are using the Wkipedia corpus, where strong association between multi-word terms can be found easily using collocation score.

Therefore, no need to add one extra multi-word terms extraction system. It will only increase the complexity of proposed method. The overall complexity of the proposed method depends on the number of senses present for each word in the sentence. Let there is N number of senses for a content word W. For each sense, we have to find a collocation extraction score. In each sense, there is M number of content words. The proposed method takes $\mathcal{O}(N \times M)$ times to compute the score for each sense. The best case of the proposed method is that when the target word has only one sense. In that scenario, no need to find the sense. In the average or worst case the complexity will be $\mathcal{O}(N \times M)$. The values of N and M will not be high. Therefore, the complexity will not be high. see Fig. 1.

9. Conclusion and future work

Ambiguity removal or detecting senses is an emerging problem in natural language processing and ontology. Sense makes an important role while finding semantic relatedness score between two content words. Ambiguous word always degrades the quality of query-based text summarization. Ambiguity in words minimizes the query relevant sentences for query-oriented text summarization. This work has presented a hybrid technique for detecting the word sense on the basis of the collocation score. Detection of correct sense disambiguates the word. The proposed method combines both a corpus-based technique and a knowledge-based technique. The experimental evaluation and discussion have been carried out using Senseval and SemEval datasets. From the above analysis and evaluation, it is clear that the proposed method works better than many existing and current systems. This word sense will contribute in calculating accurate semantic relatedness score between two words with respect to the context of the sentence. In the future, we can extend our proposed method to the multilingual system. One limitation of our work is that it can find the accurate sense of a word only if that word is present in WordNet. Hence, future work can be carried out for finding the sense for those words which are not present in WordNet. We can also add exact sense of a query word as an expanded query term and use it in many fields like information retrieval, question–answering or query-based text summarization systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agirre, E., López de Lacalle, O., Soroa, A., 2014. Random walks for knowledge-based word sense disambiguation. Comput. Linguistics 40, 57–84.
Agirre, E., Soroa, A., 2009. Personalizing pagerank for word sense disambiguation. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 33–41.
Arab, M., Jahromi, M.Z., Fakhrahmad, S.M., 2016. A graph-based approach to word sense disambiguation. an unsupervised method based on semantic relatedness. In: 2016 24th Iranian Conference on Electrical Engineering (ICEE). IEEE, pp. 250–255.
Banerjee, S., Pedersen, T., 2002. An adapted lesk algorithm for word sense disambiguation using wordnet. In: International Conference on Intelligent Text Processing and Computational Linguistics. Springer, pp. 136–145.
Banerjee, S., Pedersen, T., 2003. Extended gloss overlaps as a measure of semantic relatedness. Ijcai 3, 805–810.
Basile, P., Caputo, A., Semeraro, G., 2014. An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 1591–1600.
Bird, S., Klein, E., Loper, E., 2009. Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media Inc..
Chaplot, D.S., Bhattacharyya, P., Paranjape, A., 2015. Unsupervised word sense disambiguation using markov random field and dependency parser. In: Twenty-Ninth AAAI Conference on Artificial Intelligence.
Corrêa Jr, E.A., Amancio, D.R., 2019. Word sense induction using word embeddings and community detection in complex networks. Physica A 523, 180–190.
Denoyer, L., Gallinari, P., 2006. The wikipedia xml corpus. In: International Workshop of the Initiative for the Evaluation of XML Retrieval. Springer, pp. 12–19.
Dongsuk, O., Kwon, S., Kim, K., Ko, Y., 2018. Word sense disambiguation based on word similarity calculation using word vector representation from a knowledge-based graph. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2704–2714.
Haveliwala, T.H., 2002. Topic-sensitive pagerank. In: Proceedings of the 11th international conference on World Wide Web. ACM, pp. 517–526.
Hirst, G., St-Onge, D., et al., 1998. Lexical chains as representations of context for the detection and correction of malapropisms. WordNet: An electronic lexical database 305, 305–332.
Iacobacci, I., Pilehvar, M.T., & Navigli, R. (2016). Embeddings for word sense disambiguation: An evaluation study. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 897–907). volume 1.
Jiang, J.J., & Conrath, D.W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008.
Kågebäck, M., & Salomonsson, H. (2016). Word sense disambiguation using a bidirectional lstm. arXiv preprint arXiv:1606.03568.
Le, M., Postma, M., Urbani, J., Vossen, P., 2018. A deep dive into word sense disambiguation with lstm. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 354–365.
Leacock, C., Chodorow, M., 1998. Combining local context and wordnet similarity for word sense identification. WordNet: An electronic lexical database 49, 265–283.
Lesk, M., 1986. 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, pp. 24–26.
Li, Q., Feng, L., Pei, J., Wang, X.S., Zhou, X., Zhu, Q., 2009. Advances in Data and Web Management: Joint International Conferences, APWeb/WAIM 2009, Suzhou, China, April 2–4, 2009, Proceedings, volume 5446. Springer.
Li, Y., Bandar, Z.A., McLean, D., 2003. An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans. Knowledge Data Eng. 15, 871–882.

Martin, J.H., & Jurafsky, D. (2009). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. Pearson/Prentice Hall Upper Saddle River.
McKeown, K.R., Radev, D.R., 2000. Collocations. Handbook of Natural Language Processing. Marcel Dekker.
McRae, K., Khalkhali, S., & Hare, M. (2012). Semantic and associative relations in adolescents and young adults: Examining a tenuous dichotomy.
Melamud, O., Goldberger, J., Dagan, I., 2016. context2vec: Learning generic context embedding with bidirectional lstm. In: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pp. 51–61.
Miller, G.A., 1995. Wordnet: a lexical database for english. Commun. ACM 38, 39–41.
Miller, G.A., Chodorow, M., Landes, S., Leacock, C., Thomas, R.G., 1994. Using a semantic concordance for sense identification. In: Proceedings of the workshop on Human Language Technology. Association for Computational Linguistics, pp. 240–243.
Moro, A., Raganato, A., Navigli, R., 2014. Entity linking meets word sense disambiguation: a unified approach. Trans. Assoc. Comput. Linguistics 2, 231–244.
Navigli, R., Lapata, M., 2009. An experimental study of graph connectivity for unsupervised word sense disambiguation. IEEE Trans. Pattern Anal. Mach. Intell. 32, 678–692.
Navigli, R., Ponzetto, S.P., 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. 193, 217–250.
O'Hara, T., Bruce, R., Donner, J., Wiebe, J., 2004. Class-based collocations for word sense disambiguation. In: Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text.
Pasini, T., Navigli, R., 2017. Train-o-matic: Large-scale supervised word sense disambiguation in multiple languages without manual training data. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 78–88.
Pedersen, T., 2001. A decision tree of bigrams is an accurate predictor of word sense. In: Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies. Association for Computational Linguistics, pp. 1–8.
Perkins, J., 2014. Python 3 text processing with NLTK 3 cookbook. Packt Publishing Ltd..
Plaut, D.C. (1995). Semantic and associative priming in a distributed attractor network. In Proceedings of the 17th annual conference of the cognitive science society (pp. 37–42). Pittsburgh, PA volume 17.
Popov, A. (2017). Word sense disambiguation with recurrent neural networks. In Proceedings of the Student Research Workshop associated with RANLP (pp. 25–34).
Rada, R., Mili, H., Bicknell, E., Blettner, M., 1989. Development and application of a metric on semantic nets. IEEE Trans. Syst., Man, Cybern. 19, 17–30.
Raganato, A., Bovi, C.D., Navigli, R., 2017. Neural sequence learning models for word sense disambiguation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1156–1167.
Raganato, A., Camacho-Collados, J., & Navigli, R. (2017). Word sense disambiguation: A unified evaluation framework and empirical comparison. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers (pp. 99–110). volume 1.
Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007.
Rothe, S., & Schütze, H. (2015). Autoextend: Extending word embeddings to embeddings for synsets and lexemes. arXiv preprint arXiv:1507.01127.
Singh, R.L., Ghosh, K., Nongmeikapam, K., Bandyopadhyay, S., 2014. A decision tree based word sense disambiguation system in manipuri language. Advanced Computing 5, 17.
Sparck Jones, K., 1972. A statistical interpretation of term specificity and its application in retrieval. J. Documentation 28, 11–21.
Taghipour, K., Ng, H.T., 2015. One million sense-tagged instances for word sense disambiguation and induction. In: Proceedings of the nineteenth conference on computational natural language learning, pp. 338–344.
Tong, H., Faloutsos, C., Pan, J.-Y., 2006. Fast random walk with restart and its applications. In: Sixth International Conference on Data Mining (ICDM'06). IEEE, pp. 613–622.
Tripodi, R., Pelillo, M., 2017. A game-theoretic approach to word sense disambiguation. Comput. Linguistics 43, 31–70.
Weissenborn, D., Hennig, L., Xu, F., & Uszkoreit, H. (2015). Multi-objective optimization for the joint disambiguation of nouns and named entities. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 596–605). volume 1.
Wiriyathammabhum, P., Kijsirikul, B., Takamura, H., & Okumura, M. (2012). Applying deep belief networks to word sense disambiguation. arXiv preprint arXiv:1207.0396.
Wu, Z., Palmer, M., 1994. Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, pp. 133–138.
Yuan, D., Richardson, J., Doherty, R., Evans, C., & Altendorf, E. (2016). Semi-supervised word sense disambiguation with neural models. arXiv preprint arXiv:1603.07012.
Zhong, Z., Ng, H.T., 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In: Proceedings of the ACL 2010 system demonstrations, pp. 78–83.
Zhong, Z., Ng, H.T., 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In: Proceedings of the ACL 2010 System Demonstrations. Association for Computational Linguistics, Uppsala, Sweden, pp. 78–83.