

A Method to Disambiguate a Word by Using Restricted Boltzmann Machine

Nazreena Rahman¹ and Bhogeswar Borah²

¹Assam Kaziranga University, Jorhat-785006, India

²Tezpur University, Sonitpur-784028, India

nazreena.rehman@gmail.com, bgb@tezu.ernet.in

Abstract

To find the correct word's sense is a great importance in many textual data related applications such as information retrieval, text mining and natural language processing. We have proposed one novel Word Sense Disambiguation (WSD) method according to its context. On the Basis of collocation extraction score, three different features are extracted for each sense definition of a target word. From the extracted features, feature vector is created. A sense matrix is formed from all the feature vectors. To enhance the sense matrix, Restricted Boltzmann Machine (RBM) is used. By using SENSEVAL and Sem Eval datasets, proposed WSD method is compared with other current systems. Practical implementation of the proposed WSD method is also shown here by applying it on query-based text summary. To implement it in query-based text summary, the method uses DUC (Document Understanding Conference) datasets. It contains news-wire articles. Finally, the experimental analysis shows that our proposed WSD method outperforms many current query-based text summary systems.

1 Introduction

Disambiguation of word is much important in the fields of natural language processing and ontology. In computational linguistics, most of the languages are polysemous (Rahman and Borah, 2021b). For example, 'I am going to bank.' The bank could be a financial institution or it could be a sloping land. In a sentence, sense of a word depends on context of the sentence (Lin et al., 2016). It is very troublesome to discover the sense of a word in computer programs because it does not possess endless information like a human being. According to Jurafsky

et al. (Jurafsky, 2000), the task of selecting the correct sense of a word is known as WSD. Many text applications depend on WSD technique in their process.

The original Lesk algorithm (Lesk, 1986) uses gloss or definition of the ambiguous word (Kwon et al., 2021). Lesk algorithm can be applied only in short phrases. Wawer et al. (Wawer and Mykowiecka, 2017) developed two approaches based on the supervised and unsupervised method. The first method is an unsupervised method where log probability is computed from the sequences of word embedding vectors by considering the senses of the ambiguous word and from the context, it finds the correct sense (Rahmani et al., 2021). The second method is a supervised method where a multilayer neural network is used to find the sense of an ambiguous word. Another unsupervised method was developed for word sense disambiguation by Chaplot et al. (Chaplot and Salakhutdinov, 2018). Here, the whole text document is considered as a context for the ambiguous word. This model is based on logistic normal topic model, which adds semantic information about the synsets as its priors.

Literature survey says that different supervised, unsupervised, and knowledge-based approaches are widely used in WSD ((Navigli, 2009) (Bevilacqua et al., 2021)). A word contains different senses based on context and we need to disambiguate the target word for that given context. Word sense disambiguation method is applied in many fields like sentence similarity measure. Pawar et al. (Pawar and Mago, 2018) used 'max similarity' algorithm for WSD (Pedersen et al., 2005) where they calculated sentence similarity score. They have implemented in *Pywsd* which is available in NLTK library in Python (Tan, 2014). However, the ac-

curacy of this method is quite low as it does not always provide the exact sense of a word as maximum similarity does not give the surety that two senses will be exact for both sentences.

Here, an unsupervised learning algorithm named as Deep Belief Network (DBN) is applied. DBN is a probabilistic learning model having multi-layers of hidden units (Wiryathammabhum et al., 2012). In DBN, a Restricted Boltzmann Machine (RBM) is used to train the model. Three different features are proposed to find the exact sense of a word. Finally, RBM is used to enhance the extracted features to improve the result.

2 Contribution

The main contribution is that proposed WSD method depends on three new features based on collocation extraction scores and further Restricted Boltzmann Machine is applied to enhance these three features which give better result than other existing word sense disambiguation systems. To the best of my knowledge, there is no such kind of earlier work done in disambiguation of a word by using DBN. The proposed WSD method is the original one and can be applicable in many text mining applications. This proposed WSD method can be used in semantic relatedness score calculation between two sentences as it finds the word's correct sense on the basis of context of the sentence. Further, semantic relatedness measure can be applied in many text mining applications like query-based text summarization, text clustering, plagiarism detection. The proposed WSD method is applied to query-based text summarization datasets to show its practical implementation. Query-based text summarization is different from generic summarization as it extracts essential sentences from the input text based on the user's requirement (Rahman and Borah, 2020). Therefore, to find semantic relatedness between query and input text sentence, this WSD technique is applied to get accurate relatedness score.

3 Introduction to WordNet

WordNet (Miller, 1995) is a lexical dictionary. Only content words are present in WordNet. These words are organized semantically. It is different from the traditional dictionary. Nouns, verbs, adverbs, and adjectives are present in content words. 'WordNet' contains synonymous words set. It is known as *synset* or *synonym set*. Synonym set

contains words having same meaning. For example, *shut* and *close* are synonyms. Polysemous words possess more than one *synsets*. For example, *right*; sometimes it means correct, morally good or justified and sometimes used as direction opposite to left. For each content word present in a synset, a gloss or a definition is present. Most of the content words contain more than one sense definition. In Wordnet, a word is represented as *word#part_of_speech#sense_number*. Table 1 says about different gloss definition of word *love* along with its parts-of-speech and sense number present in WordNet.

WordNet dictionary is used here for calculating the semantic similarity or relatedness score between two words. Therefore, before finding the score, it is important to disambiguate those words.

4 Proposed Unsupervised Deep Learning Method for Sense Detection

Process of finding the appropriate sense of a word is shown in Figure 1. Following steps are used to find the correct word's sense present in a sentence using unsupervised deep learning:

- Pre-processing: Initially, pre-processing step removes unwanted words from the text sentence. It makes the sentence a lighter one. Here, pre-processing of text document uses stop word removal. Stop words eliminates most common and unimportant words. Example of stop word are: *are, is, the* etc.
- Feature Extraction: For finding sense of a word, three features are used. It is described in section 5 and 6.
- Feature enhancement: Feature enhancement is done to improve the selection of sense for the context of the sentence. RBM is used for a feature enhancement to get the exact sense. How RBM can be applied in feature enhancement for finding word sense is described in section 7.

5 Finding Collocation Extraction Score between two Words

Collocation refers as the use or occurrence of two words together. Computational technique to find the collocation in a document or a corpus is known

Table 1: Representing content ‘love’ in WordNet

Synset(‘love.n.01’)	a strong positive emotion of regard and affection
Synset(‘love.n.02’)	any object of warm affection or devotion
Synset(‘love.n.03’)	a beloved person; used as terms of endearment
Synset(‘love.n.04’)	a deep feeling of sexual desire and attraction
synset (love.n.05’)	a score of zero in tennis or squash
Synset(‘love.v.01’)	have a great affection or liking for
Synset(‘love.v.02’)	get pleasure from
Synset(‘love.v.03’)	be enamored or in love with

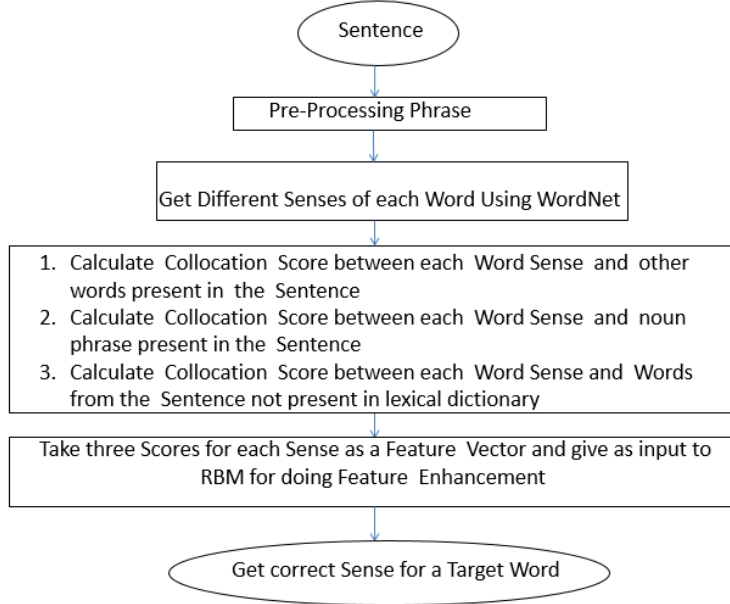


Figure 1: Block Diagram of Word Sense Disambiguation Method

as collocation extraction score (Rahman and Borah, 2021a). To find collocation extraction score, Wikipedia Corpus (WC) (Denoyer and Gallinari, 2006) is used. Bi-gram frequency is used to find the co-occurrence between two terms. Associativity is found in while calculating collocation. If we take the example of cat and tiger, both are semantically similar. Both are members of the feline family or superb hunters. In contrast, tiger and deer are associated as both occur frequently in language. This is known as functional relationship. Association and similarity both are not even mutually exclusive or independent. Two words tiger and deer are related two both relations to some degree (McRae et al., 2012) (Plaut, 1995). For each sense of ambiguous word, bi-gram collocation extraction score $word_1$ (McKeown and Radev, 2000) is found by calculating the frequency of words’ available in sense definition for the first word ($word_1$) with the present words in the sentence and finally maximum value is taken by the proposed WSD method.

We have taken one sentence: *Ram went to the*

state bank of India for depositing money. Initially, for fining the accurate sense of word *bank*, we obtain all the senses present in WordNet. We find the collocation extraction score for each word present with the sense with the other content words present in the sentence. There are many senses present for the word *bank*. For example if I take the sense *depository financial institution*, then for each content word present in the sense *deposit, financial, institution*, we need to find the collocation extraction score with the content words present in the sentence *Ram, Go, State, India, deposit, money*. At the end, we need to take the maximum value. In this way we have to find the score for each sense. Following equation 1 is used for finding the collocation extraction score between two words. One word is selected from the gloss of g and other word is selected from the sentence sen :

$$collocation_score(g, sen) = \frac{\log\left(\frac{(z*SC)}{(gf*sf*span)}\right)}{\log(2)} \quad (1)$$

gf = frequency of g in WC
sf = frequency of sen in WC
z = frequency of sen near g in WC
SC= size of WC
span = width of the words (e.g. 2 to left and 2 to right of first word)

For a target word (TaW) exist in a sentence, the collocation extraction score (CES) of that sense is:

$$CES(Sense, Sentence) = \max_{g \in Sense, sen \in Sentence} (collocation\ score(g, sen)) \quad (2)$$

We find the collocation extraction score for all the senses of TaW . Finally, we select that sense of TaW for which the proposed WSD method gets the maximum collocation extraction score (Rahman and Borah, 2021b).

6 Feature Extraction for Finding Exact Sense of a Word present in a Sentence

Initially, proposed WSD method takes all the content words available in the same sentence to detect the sense of an ambiguous word. For an ambiguous word, the proposed method takes all senses present in the WordNet. For each sense, collocation extraction score is calculated for all content words. To find out the collocation extraction score, algorithm 1 shows the systematic steps:

Data: target word (T_w) and the sentence (S_{ws})

Result: collocation_score of T_w for each sense s of T_w

Do the stop word removal of S_{ws}

Find out senses of T_w

for each sense (s) of T_w do

 Do the stop word removal (s_{swr}) of s

for each word (w_{swr}) of s_{swr} do

 Find out the collocation extraction score between w_{swr} and S_{ws} by using the equation 2

end

end

Algorithm 1: Collocation Extraction Score for target word's each sense with words available in the sentence

It is also seen that noun phrases always carry essential information which helps in finding the

meaning of a sentence. Therefore, noun phrases are considered and find collocation extraction score. We have calculated the collocation extraction score for each sense of target word with noun phrases. Description of Algorithm 2 is given below.

Data: target word (T_w) and sentence (S_{ws})

Result: collocation_score of T_w for each sense s of T_w

Do the removal of stop words S_{ws}

Find the noun phrases (S_{np}) in S_{ws}

Find out senses of T_w

for each sense (s) in T_w do

 Do the stop word removal (s_{swr}) of s

for each word (w_{swr}) of s_{swr} do

 Find out the collocation extraction score between w_{swr} and S_{np} by using the equation 2

end

end

Algorithm 2: Collocation Extraction Score of each sense of the target word with the noun phrases present in the sentence

It is also observed that sometimes, some words are not present in the WordNet, but they still can be considered as important words as they contribute to creating the context of the sentence. For example, we take three sentences: *Narendra Modi visits China*, *Shyam visited his uncle's house to attend the birthday party*, and *Donald Trump visited India*. The word *Visit* has different senses in WordNet. Now the proposed method will find the most suitable sense present in WordNet. In first sentence, the word *visit* is much related to *Narendra Modi*. It can understand that this *visit* must be an official visit, as Prime Minister usually goes to other foreign countries for official purpose. The same meaning is also present for the third sentence. For the second sentence, the word *visit* is related to go to see a place which is certainly not official. To find exact sense, the collocation extraction score between each sense of *visit* with *Narendra Modi*, *Shyam*, and *Donald Trump* will contribute to it. *Narendra Modi*, *Donald Trump*, and *Shyam* are not present in WordNet. That refers that, sometimes, if a word is not present in WordNet, still it helps in contributing to finding exact sense. The word *visit* is identified as a verb here.

From the above Table ??, it is clear that the sense of *visit* word for the first and third sentence should

be visit#v#4 as it matches with the context of the sentence and for the second sentence, it should be visit#v#1. The collocation extraction score between *Narendra Modi* and *office* is much higher in case of the first sentence; for the second sentence it is for *party* and *entertainment* and the third sentence it is for *Donald Trump* and *office*. This exact sense will help in finding an exact semantic relatedness score among three sentences. Therefore, the following Algorithm 3 is used to find the required collocation extraction score:

Data: target word (T_w) and the sentence (S_{ws})
Result: collocation_score of T_w for each sense s of S_{ws}
Do stop word removal of T_w
Find the words from S_{ws} not present in WordNet (S_{wsnw})
Find out senses of T_w
for each sense (s) in T_w **do**
 Do the stop word removal (s_{swr}) of s
 for each word (w_{swr}) of s_{swr} **do**
 Find collocation extraction score between w_{swr} and S_{wsnw} with the help of equation 2
 end
end

Algorithm 3: Collocation Extraction Score of target word with words exist in the sentence but not present in WordNet

7 Use of Restricted Boltzmann Machine for Feature Enhancement to Find Correct Sense

We use Restricted Boltzmann Machine (RBM) for finding correct sense. It needs the collocation extraction score of each target word obtained from the mentioned Algorithm1, Algorithm 2, and Algorithm 3. For each target word, every sense is presented as a feature vector. For example: if a target word w has n number of senses, then it is represented as a feature vector: $w_1 = [f_{11}, f_{12}, f_{13}]$, $w_2 = [f_{21}, f_{22}, f_{23}]$, $w_n = [f_{n1}, f_{n2}, f_{n3}]$. This sense matrix will be input for the RBM. The feature vector for each target word is passed through the hidden layer where each feature vector is multiplied with respective weight and a bias value.

Initially, train the RBM with the input vector v .

Now, the hidden units of RBM become deterministic. RBM has two layers: one is visible, or input layer, and the other is a hidden layer. No of units in the visible layer depends on how many senses are present for each word available in the text document. It calculates $E(h_j|v; w)$ with input v which serves again as visible units for RBM. This process can be repeated as many layers as it needs. After this unsupervised layer-wise training, back propagation is utilized to fine-tune of weights and biases (Cai et al., 2012). This unsupervised phase does not require labels. Finally, we will get a refined and enhanced matrix.

The method can be represented mathematically. Input for the RBM is a sense matrix. Each row in the matrix represents one particular sense of a word. Whole content words present in the text document are represented as a sense matrix. The sense matrix $S = (s_1, s_2, \dots, s_N)$ is a feature vector set contains all the three features extracted for each sense of a word s_1 (Jain and Lobiyal, 2022). The sense matrix is represented by the following Figure 2:

$$\begin{matrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{matrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ f_{N1} & f_{N2} & f_{N3} \end{bmatrix}$$

Figure 2: Sense Matrix

The input to the RBM is the set of feature vector S . It acts as a visible layer. Here, random values are selected for biases. RBM contains two hidden layers, the whole process can be represented in following equations:

$$S = (s_1, s_2, \dots, s_N) \quad (3)$$

Here $s_i = (f_1, f_2, f_3)$ and $i \leq N$, where N is the sense number for the ambiguous word present in the text document. As the RBM has two layers, two bias values h_0 and h_1 are selected randomly.

To get a more refined matrix, RBM works in two steps. During the first phase, the new refined sense matrix is:

$$S' = (s'_1, s'_2, \dots, s'_N) \quad (4)$$

The above expression is obtained in the follow-

ing way:

$$\sum_{i=1}^N s_i + h_0 \quad (5)$$

In step 2, the same procedure is followed by considering the bias h_1 and get more enhanced and refined sense matrix and which is given by:

$$S'' = (s''_1, s''_2, \dots, s''_N) \quad (6)$$

After obtaining refined sense matrix, a threshold value is taken for testing purpose with it. The threshold value is randomly generated for each vector. it is further tested with a For example: if the value of $f_1 \geq threshold_{f_1}$, then only it will be considered.

To generate optimal feature vector set, obtained feature vector sets are fine tuned by adjusting the units' weight of RBM. For optimal fine-tuning, back propagation algorithm is used. The enhanced feature vector values are added to obtain a score against each sense. Finally, the highest scored sense will be considered as the best sense for that target word. Note that the RBM will have to be freshly trained for each new content word that has to be disambiguated.

8 Experimental Analysis and Discussion

8.1 Evaluation Metric

For evaluation of the proposed word sense disambiguation method, the following equation 7 calculates the performance of the method.

$$mi = \frac{\text{correctly predicted instances number}}{\text{all test instances number}} \quad (7)$$

This mi stands for micro-average recall (Wiriyathammabhum et al., 2012).

8.2 Experiment with Word Sense Disambiguation Datasets

Proposed WSD Method is compared with other recognized and current word sense disambiguation methods where SENSEVAL-2, Sem Eval-2013 task 12 and SemEval-2015 task 13 datasets are used ((Ide and Véronis, 1998), (Wiriyathammabhum et al., 2012),(Navigli et al., 2013)). Three different features: topical local and part-of-speech- are used by Wiriyathammabhum et al. They have used different learning methods on SENSEVAL-2 dataset for word sense disambiguation (Wang et al., 2017).

MFS (Most Frequent Sense) method is considered as a baseline method which chooses the major class of each word task as its prediction. Table 2 shows that proposed WSD Method performs better than existing methods. Though RBM is used by Wiriyathammabhum et al. (Wiriyathammabhum et al., 2012), but their used features are dissimilar from the proposed WSD method. Therefore, the result varies, and the proposed method performs well.

Table 2: Micro-average recall values of various learning algorithms with proposed WSD Method

Method Name	Accuracy in percentage (mi)
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 WSD	72.80%

Unsupervised and supervised BabelNet-based WSD systems are used for comparison purpose (Dongsuk et al., 2018). F-score is used here as evaluation metric. BabelNet is a multilingual encyclopedic dictionary (Navigli and Ponzetto, 2012). Following widely used unsupervised systems: Moro et al. (Moro et al., 2014), Agirre et al. (Agirre et al., 2014), Apidianaki et al. (Apidianaki and Gong, 2015), Tripodi et al. (Tripodi and Pelillo, 2017), Dongsuk et al. (Dongsuk et al., 2018) and supervised systems: Zhong et al. (Zhong and Ng, 2010), Weissenborn et al. (Weissenborn et al., 2015), Raganato et al. (Raganato et al., 2017), Pasini et al. (Pasini and Navigli, 2017) are considered here. For SemEval-2013 dataset, The performance of our proposed WSD method is better than existing WSD systems. It is also seen in Table 3 that although for SemEval-2015 dataset, supervised Weissenborn et al. method performs better than our proposed WSD method but for macro-average score, proposed WSD method has shown better performance. A macro-average takes the average value of F-scores.

In comparison to other existing unsupervised (knowledge-based) methods, the proposed WSD method shows better performance. Though performance of some supervised methods are better than the existing knowledge-based methods ((Raganato et al., 2017)), but literature survey says that

it is quite expensive to construct the training corpus for all the languages and words. Hence, this is one of the prominent limitation of supervised approach while applying in WSD. On other hand, WordNet ((Banerjee and Pedersen, 2003), (Chaplot et al., 2015)) is used in knowledge-based WSD system. In knowledge-based WSD systems, contextual information and semantic knowledge both are incorporated. Therefore, knowledge-based approach can disambiguate larger number of words. Conclusion can be derived from this discussion is that WSD systems which are based knowledge are more practicable and attainable than supervised WSD systems ((Chaplot et al., 2015), (Moro et al., 2014), (Chaplot and Salakhutdinov, 2018), (Dongsuk et al., 2018)). It is also tested that if anyone feature is dropped from the three proposed features, the overall performance degrades. It can be said that all three features are equally important.

9 Comparison of Results of use of Three different Features with and without the Use of Feature Enhancement

Here, we have compared the word sense disambiguation results with or without the use of feature enhancement technique. Through this comparison, it is quite clear that the feature enhancement through Restricted Boltzmann Machine helps in getting better results. Following Table 4 shows the performance of word sense disambiguation method with or without using feature enhancement in Sem Eval datasets. From the comparison 4, it is quite clear that there is a high impact of utilization of RBM in disambiguation of sense of a word.

9.1 Evaluation on Query-Based Text Summarization Datasets

9.1.1 Datasets

To further prove the performance of the proposed WSD method in practical implementations, evaluation is done on query-based text summarization datasets. WSD is widely used in text summarization. WSD helps in extracting more query oriented sentences for creating query-based text summarization. Newswire articles are taken from the Document Understanding Conference (DUC) corpora to implement WSD method. Effectiveness of the proposed method is evaluated with existing systems that perform an experimental evaluation using the Document Understanding Conference. DUC 2005 and 2006 datasets (<http://duc.nist.gov>) are mainly

used in query-based text summarization purpose (Gervasi et al., 2019). They have complex real-life query with related text documents. Datasets contain 50 queries with 50 different topics and length of the summary is of 250 words only.

9.1.2 Evaluation of Proposed WSD Method with DUC 2005 and DUC 2006 Systems

At first, proposed WSD methods is compared with DUC 2005 and 2006 datasets. Sense of each content word is found. Table 5 and Table 6 present the different *mi* scores for DUC 2005 and 2006 datasets. The proposed WSD method is compared with the baseline method, along with some other existing and widely used WSD methods. Here, the baseline system represents the LESK algorithm. Table 5 and Table 6 provide the scores for different *mi* values. Results show that the proposed WSD method has better performane than all the existing WSD methods.

9.1.3 Evaluation of Proposed WSD Method on Query-Based Text Summarization

Now proposed WSD method is implemented for finding query-based text summary. Commonly used HSO semantic relatedness measure ((Pedersen et al., 2004), (Hirst et al., 1998)) is applied here for calculating the semantic relatedness score between query and input text sentences. Sentences are extracted based on its semantic relatedness score. The equation to find Semantic Relatedness value (*S*) between two words *w1* and *w2* is:

$$S((w1, s_1, p_1), (w2, s_2, p_2)) = 2 \times c - PL(w1, w2) - k \times DC(w1, w2) \quad (8)$$

here,

s_1= sense number of *W1*
p_1= part of speech of *W1*
s_2= sense number of *W2*
p_2= part of speech of *W2*
PL= Path Length
DC= Direction Change

Here, values of *A* and *C* are 8 and 1, respectively. The maximum value of HSO score is 16 which means two content words are same. The minimum value of HSO score is 0 which means there is no relatedness between two content words (Xia et al., 2019). For finding the Semantic Relatedness Score (*S*) between two sentences *s1* and *s2*, we use the

Table 3: Comparison of different BabelNet-based unsupervised and current supervised 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_liter _{SRP2vSim} 18	75.0	65.8	70.4
	Proposed WSD	75.6	74.8	75.2
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 4: Comparison results with and without the use of feature enhancement

System	F-score for SemEval-13	F-score for SemEval-15	Macro Avg F-score
Proposed WSD (without Future Enhancement)	70.6	69.3	68.6
Proposed WSD (with Future Enhancement)	75.6	74.8	75.2

Table 5: mi values for different WSD methods on DUC 2005 datasets

Method Name	Accuracy in percentage (mi)
Proposed WSD	79.2%
Original_Lesk	55%
Adapted_Lesk	60%
Cosine_Lesk	61.4%

Table 6: mi values for different WSD methods on DUC 2006 datasets

Method Name	Accuracy in percentage (mi)
Proposed WSD	81.20%
Original_Lesk	57%
Adapted_Lesk	64.34%
Cosine_Lesk	64.42%

following equation 9:

$$S(s_1, s_2) = \frac{\sum_{w_1 \in s_1, w_2 \in s_2} S((w_1, s_{-1}, p_{-1}), (w_2, s_{-2}, p_{-2}))}{\text{Maximum relatedness score}} \quad (9)$$

Mentioned existing WSD methods are considered again and now we use that appropriate sense for calculating S between query and input text sentences to create query-based text summary. We have taken the threshold value as

60%. It means that input sentences which are equal or greater than the threshold value are all equally important for query-based text summary creation. For comparison purpose, length of the summary is confined to 250 words for DUC datasets. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) ((Lin, 2004)) is used here for evaluation purpose. ROUGE is a popular and standard intrinsic-based metric. National Institute for Standards and Technology (NIST) adapts ROUGE for summarization evaluation metric. To compare different summaries, different metrics are available in ROUGE. Quality of summary is measured in terms of overlapping units such as N-grams, word sequences, and word pairs. Different ROUGE measures ROUGE-N (N-gram co-occurrence), ROUGE-L (Longest Common Subsequence), ROUGE-W (Weighted Longest Common Subsequence), ROUGE-S (Skip-Bigram) and ROUGE-SU: (Extension of ROUGE-S) are available. ROUGE-N is a $gram_n$ recall between system-generated summary and human summary. It is based on the total number of common content words between them. Equation to find ROUGE-N is :

$$ROUGE - N = \frac{\sum_{S \in H_S} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in H_S} \sum_{gram_n \in S} Count(gram_n)}$$

Here, N is the length of $gram_n$.

$Count_{match}(gram_n)$ says about the total common $grams_n$ co-occurring in both system and human summary and $Count(gram_n)$ gives the number of $grams_n$ present in human summary. Here, official metrics of ROUGE-1, ROUGE-2 and ROUGE-SU4 are used along with 95% confidence intervals. Tables 7 and 8 present different ROUGE scores.

Table 7: Different ROUGE values for Query-Based Text Summary on DUC 2005 datasets

Method Name	ROUGE-1	ROUGE-2	ROUGE-SU4
Proposed WSD	0.3812	0.0752	0.1413
Original_Lesk	0.3711	0.0624	0.1218
Adapted_Lesk	0.3768	0.0651	0.1291
Cosine_Lesk	0.3791	0.0687	0.1317

Table 8: Different ROUGE values for Query-Based Text Summary on DUC 2006 datasets

Method Name	ROUGE-1	ROUGE-2	ROUGE-SU4
Proposed WSD	0.4017	0.0921	0.1482
Original_Lesk	0.3992	0.0861	0.1479
Adapted_Lesk	0.4001	0.0891	0.1489
Cosine_Lesk	0.4009	0.0897	0.1494

From the evaluations, it is quite clear that the proposed WSD method helps in getting more query relevance sentences, which helps in creating a query-based text summary. Here, only the semantic relatedness measure is used. In the future, features can be increased, which will help in extracting more query related sentences.

10 Conclusion and Future Work

We have presented an unsupervised deep learning technique for detecting the word’s sense. Three different features are extracted based on the collocation score. Restricted Boltzmann Machine is used to enhance the features. Proposed WSD method is implemented on word sense disambiguation datasets to compare mainly with other existing and current word sense disambiguation methods. The evaluation shows that the proposed WSD method outperforms many current methods. As this method will be used in query-based text summarization, evaluations have done on DUC datasets where the performance is much more better than many

query-based text summarization methods. Experimental analysis shows better performance of our proposed WSD method than many current methods. The result attains much better due to the use of collocation based features in deep belief network. In future, the proposed WSD method can be used in many fields like question-answering, information retrieval or query-based text summarization. The proposed method will also try to work on languages other than English.

References

- Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.
- Marianna Apidianaki and Li Gong. 2015. Limsi: translations as source of indirect supervision for multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 298–302.
- Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 805–810. Acapulco, Mexico.
- Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, Roberto Navigli, et al. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conference on Artificial Intelligence, Inc.
- Xianggao Cai, Su Hu, and Xiaola Lin. 2012. Feature extraction using restricted boltzmann machine for stock price prediction. In *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, volume 3, pages 80–83. IEEE.
- Devendra Singh Chaplot, Pushpak Bhattacharyya, and Ashwin Paranjape. 2015. Unsupervised word sense disambiguation using markov random field and dependency parser. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Devendra Singh Chaplot and Ruslan Salakhutdinov. 2018. Knowledge-based word sense disambiguation using topic models. *arXiv preprint arXiv:1801.01900*.
- Ludovic Denoyer and Patrick Gallinari. 2006. The wikipedia xml corpus. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 12–19. Springer.

- O Dongsuk, Sunjae Kwon, Kyungsun Kim, and Youngjoong Ko. 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*, pages 2704–2714.
- Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Giuseppe Borruso, Carmelo M Torre, Ana Maria AC Rocha, David Taniar, Bernady O Apduhan, Elena Stankova, and Alfredo Cuzzocrea. 2019. Computational science and its applications–iccsa 2017. In *ICCSA (Conference)*, 1, pages 1–1. Springer,.
- Graeme Hirst, David St-Onge, et al. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In *C. Fellbaum (Ed.), WordNet: An electronic lexical database*, chapter 13:305–332. Cambridge, Mass.: MIT Press.
- Nancy Ide and Jean Véronis. 1998. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics*, 24(1):2–40.
- Goonjan Jain and DK Lobiyal. 2022. Word sense disambiguation using cooperative game theory and fuzzy hindi wordnet based on conceptnet. *Transactions on Asian and Low-Resource Language Information Processing*, 21(4):1–25.
- Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.
- Sunjae Kwon, Dongsuk Oh, and Youngjoong Ko. 2021. Word sense disambiguation based on context selection using knowledge-based word similarity. *Information Processing & Management*, 58(4):102551.
- Michael Lesk. 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*, pages 24–26. ACM.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.
- Chin-Yew Lin, Nianwen Xue, Dongyan Zhao, Xuanjing Huang, and Yansong Feng. 2016. *Natural Language Understanding and Intelligent Applications: 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and 24th International Conference on Computer Processing of Oriental Languages, ICCPOL 2016, Kunming, China, December 2–6, 2016, Proceedings*, volume 10102. Springer.
- Kathleen R McKeown and Dragomir R Radev. 2000. Collocations. In: Dale, R., Moisl, H., Somers, H. (Eds.), *Handbook of Natural Language Processing*, chapter 21. Marcel Dekker, New York. Citeseer.
- Ken McRae, Saman Khalkhali, and Mary Hare. 2012. Semantic and associative relations in adolescents and young adults: Examining a tenuous dichotomy. In *In V. F. Reyna, S. B. Chapman, M. R. Dougherty, & J. Confrey (Eds.), The Adolescent Brain: Learning, Reasoning, and Decision Making*, pages 39–66. Washington, DC: American Psychological Association (APA).
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.
- Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.
- Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 222–231.
- Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.
- Tommaso Pasini and Roberto Navigli. 2017. Trainomatic: 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*, pages 78–88.
- Atish Pawar and Vijay Mago. 2018. Calculating the similarity between words and sentences using a lexical database and corpus statistics. *arXiv preprint arXiv:1802.05667*.
- Ted Pedersen, Satanjeev Banerjee, and Siddharth Patwardhan. 2005. Maximizing semantic relatedness to perform word sense disambiguation. Technical report, Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute.
- Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics.
- David C Plaut. 1995. Semantic and associative priming in a distributed attractor network. In *Proceedings of the 17th annual conference of the cognitive science society*, volume 17, pages 37–42. Pittsburgh, PA.

- Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 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*, volume 1, pages 99–110.
- Nazreena Rahman and Bhogeswar Borah. 2020. Improvement of query-based text summarization using word sense disambiguation. *Complex & Intelligent Systems*, 6(1):75–85.
- Nazreena Rahman and Bhogeswar Borah. 2021a. Redundancy removal method for multi-document query-based text summarization. In *2021 International Symposium on Electrical, Electronics and Information Engineering*, pages 568–574.
- Nazreena Rahman and Bhogeswar Borah. 2021b. An unsupervised method for word sense disambiguation. *Journal of King Saud University-Computer and Information Sciences*.
- Saeed Rahmani, Seyed Mostafa Fakhrahmad, and Mohammad Hadi Sadreddini. 2021. Co-occurrence graph-based context adaptation: a new unsupervised approach to word sense disambiguation. *Digital Scholarship in the Humanities*, 36(2):449–471.
- L Tan. 2014. ”pywsd: Python implementations of word sense disambiguation (wsd) technologies [software],” [online]. available: <https://github.com/alvations/pywsd>.
- Rocco Tripodi and Marcello Pelillo. 2017. A game-theoretic approach to word sense disambiguation. *Computational Linguistics*, 43(1):31–70.
- Tinghua Wang, Wei Li, Fulai Liu, and Jialin Hua. 2017. Sprinkled semantic diffusion kernel for word sense disambiguation. *Engineering applications of artificial intelligence*, 64:43–51.
- Aleksander Wawer and Agnieszka Mykowiecka. 2017. Supervised and unsupervised word sense disambiguation on word embedding vectors of unambiguous synonyms. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 120–125.
- Dirk Weissenborn, Leonhard Hennig, Feiyu Xu, and Hans Uszkoreit. 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)*, volume 1, pages 596–605.
- Peratham Wiriathamabhum, Boonserm Kijirikul, Hiroya Takamura, and Manabu Okumura. 2012. Applying deep belief networks to word sense disambiguation. *arXiv preprint arXiv:1207.0396*.
- Wenlong Xia, Qingdang Meng, Qingchuan Tao, and Ray T Chen. 2019. Non-orthogonal multiple access without channel state information for similar channel conditions. *Electronics letters*, 55(8):493–495.
- Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. *Proceedings of the ACL 2010 system demonstrations*, pages 78–83.