Natural Language Engineering (2020), **26**, pp. 413–432
doi:10.1017/S1351324919000421

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

Word sense disambiguation using implicit information

Goonjan Jain[1]* and D.K. Lobiyal[2]

[1]Department of Applied Mathematics, Delhi Technological University, New Delhi 110042, India and [2]School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India
*Corresponding author. Email: goonjan_jain@hotmail.com

(Received 17 October 2017; revised 21 June 2019; accepted 23 June 2019; first published online 13 September 2019)

Abstract

Humans proficiently interpret the true sense of an ambiguous word by establishing association among words in a sentence. The complete sense of text is also based on implicit information, which is not explicitly mentioned. The absence of this implicit information is a significant problem for a computer program that attempts to determine the correct sense of ambiguous words. In this paper, we propose a novel method to uncover the implicit information that links the words of a sentence. We reveal this implicit information using a graph, which is then used to disambiguate the ambiguous word. The experiments show that the proposed algorithm interprets the correct sense for both homonyms and polysemous words. Our proposed algorithm has performed better than the approaches presented in the SemEval-2013 task for word sense disambiguation and has shown an accuracy of 79.6 percent, which is 2.5 percent better than the best unsupervised approach in SemEval-2007.

Keywords: Word sense disambiguation; Common sense; ConceptNet; Implicit information; Lexical priming

1. Introduction

The primary objective of word sense disambiguation (WSD) is to identify the correct sense of an ambiguous word based on the context it is used in Navigli (2009). WSD is critical for the accuracy and reliability of Natural Language Processing (NLP) tasks, such as Machine Translation (Vickrey *et al.* 2005); Carpuat and Wu (2005; 2007); (Chan, Ng, and Chiang 2007), Text Understanding (Kilgarriff 1997), Sentiment Analysis (Rentoumi *et. al.* 2009; Sumanth and Inkpen 2015; Hung and Chen 2016), Information Retrieval (Zhong and Ng 2012; Schadd and Roos 2015). Since the 1950s, many approaches have been proposed to assign senses to words (Weaver 1955).

Two of the main methodological approaches for assigning senses to words are the supervised and unsupervised methods. Supervised methods use semantically annotated corpora to train the system that is used to assign a correct sense to a word in context (Tratz *et al.* 2007; Zhong and Ng 2010; Wang, Rao, and Hu 2014). In contrast, unsupervised methods abstain from training and instead work directly on raw unannotated corpora (Mihalcea 2005; McCarthy *et al.* 2007; Tripodi and Pelillo 2017). In this paper, we propose an unsupervised approach.

Ambiguous words can be categorized as homonyms or polysemous words (Panman 1982). For example, consider the following usage of *bass* in different contexts.

1. I play *bass*$_1$ in a jazz band.
2. He was hungry and cooked *bass*$_2$.
3. The track has a wonderful *bass*$_3$ line.

The contexts of *bass*$_1$ and *bass*$_2$ are entirely different, and thus they are homonyms. However, *bass*$_1$ and *bass*$_3$ are semantically related and therefore are polysemous. While it is quite challenging

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

to disambiguate polysemous words, the proposed method disambiguates both homonyms and polysemous words.

Often the meaning of written and verbal communication is based on explicit and implicit information. Explicit information is something that is said or written, whereas implicit information is something that is meant but not said or written. Our brains process implicit information with precision to determine the correct meaning of the communication. For the examples given above, we can use common sense to understand that one cannot eat a musical instrument to satisfy hunger and similarly we cannot make music with a fish in hand. However, existing WSD techniques are unable to process this implicit information. There are many hurdles in doing this, one being a lack of contextual knowledge. The context of a word constitutes not only the words that surround the ambiguous word but also words that are not mentioned explicitly (Dunmore 1989; Fengning 1994). Taking both the word meaning and contextual information into account can efficiently solve the WSD problem.

To understand the meaning of a sentence, we use both words of the sentence and common sense. With common sense, we establish a relationship between the words, for example, "John booked a *table* for dinner." When we read this sentence, we infer several pieces of information, such as John had dinner in some restaurant, he made a prior booking, John is a person, he feels hungry and eats food. With this information, it becomes easier to perform WSD of an ambiguous word.

The approach proposed in this paper is based on a theory called lexical priming (Hoey 2005). This theory connects the lexical aspect of Corpus Linguistics and the priming aspect of Psycholinguistics. Priming takes place if exposure to one stimulus significantly increases the response to another stimulus. For example, *theatre* is recognized more quickly following *movie* than following *umbrella* (Wettler, Rapp, and Sedlmeier 2005). Lexical priming is based on the concept of lexical co-occurrence, which states that words are used along with other words and not in isolation. Every usage of a word is loaded with the context and co-texts in which it is encountered. Thus, our knowledge of a word includes the fact that it co-occurs with certain other words and in certain kinds of contexts (Hoey 2005). The validity and utility of this theory have been discussed in the literature (Pace-Sigge 2013; Rapp and Zock 2014). This theory guides the proposed algorithm, where we understand the meaning of the text by following links between the words of a sentence. Words may be linked directly or indirectly via previously attained knowledge. In this paper, we propose a novel algorithm to identify this attained knowledge that genuinely connects the words of a sentence.

We need an association network that contains knowledge about associations among words to implement the proposed algorithm. There are two methods to obtain such knowledge. One method is to ask people what a given term (*cat*) evokes in their mind (*dog, mouse,* etc.). Another method is to look at word co-occurrences in corpora and derive the associations from them (which, strictly speaking, pre-supposes that the human brain is also doing this). The proposed algorithm uses ConceptNet (Havasi, Speer, and Alonso 2007), which has gathered implicit pieces of knowledge using both of the methods mentioned above. ConceptNet was developed as a network of common sense knowledge, and is a semantic network containing information about the real world that computers should know for understanding written text.

1.1 ConceptNet

ConceptNet is a multilingual knowledge base in which words and phrases are related to one another via common sense. The knowledge contained in ConceptNet has been collected from a variety of resources, including crowdsourced resources, such as Wiktionary[a] and Open Mind Common Sense (OMCS) (Singh *et al.* 2002); purpose-driven games, such as Verbosity (von Ahn,

[a] https://en.wiktionary.org/wiki/Wiktionary:Main_Page.

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

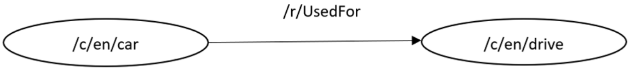

Figure 1. Single edge of ConceptNet.

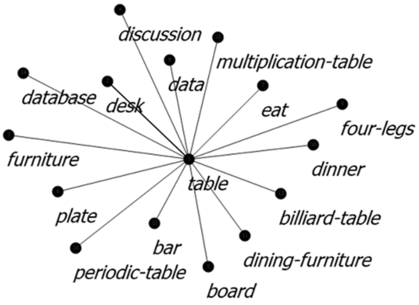

Figure 2. Example subgraph of ConceptNet with “Table” as an ambiguous word.

Keida, and Blum 2006); and expert-created resources, such as WordNet (Miller 1995; Fellbaum 1998). ConceptNet is a network of labeled nodes and edges. The nodes, or concepts, are words, word senses, and short phrases in a variety of different languages. These concepts are connected with a particular relation, which is represented as an edge. The proposition expressed by a relation between two concepts is called an assertion. Every object in ConceptNet has a uniform resource identifier (URI), which is structured similar to a path. URIs are meaningful and we can tell what the object is by looking at its URI. Different kind of objects are distinguished by the first element of the path, for example,

/c/: concept. For example, concept “car” in English has the URI “/c/en/car”. Each concept has at least three components. The initial “/c” tells that the object is a concept. The next part tells the concept’s language. Last part is the concept text.
/r/: language independent relations. For example, /r/IsA, /r/RelatedTo etc.
/a/: assertion, also known as edges. For example, the sentence “a car is used for driving” expresses an assertion, where the relation /r/UsedFor connects the concept /c/en/car to the concept /c/en/drive. The assertion has the URI: /a/[/r/UsedFor/, /c/en/car/, /c/en/drive/]

Figure 1 shows a single edge connecting two nodes where strings /c/en/car, /c/en/drive, and /r/UsedFor are URIs from ConceptNet. Figure 2 illustrates a subgraph of ConceptNet with *Table* as an ambiguous word. The *table* is connected to words belonging to different senses of its usage. The *table* is connected to *furniture, periodic table, eat* and *discussion*, which belong to different senses of the word.

“ConceptNet has no provision of representing ambiguous words or different senses of a word. It is simply a huge collection of words and phrases that are related to each other. Word senses are only distinguished by their part of speech.”[b] For example, for a word *book*, there are 11 different senses of the word as a noun and 4 different senses of it as a verb in the WordNet. However, in ConceptNet we have only two different concepts pertaining to different senses of *book*, one as a verb and the other as a noun. Their URIs are “/c/en/book/v” and “/c/en/book/n”, respectively. Each of these concepts is related to different concepts in ConceptNet.

In this paper, we have developed an algorithm for WSD, where a correct sense of an ambiguous word is suggested based on the similarity between the context of the ambiguous word, deduced using ConceptNet, and every one of its possible word senses.

The remainder of the paper is structured as follows: Section 2 gives a glimpse of some of the approaches proposed for unsupervised WSD in recent years. The detailed design and implementation methodology of our proposed work are explained in Section 3. The results of the evaluation

[b] https://github.com/commonsense/conceptnet5/wiki/FAQ

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

of the proposed algorithm by undertaking various experiments and its comparison with other available algorithms are reported in Section 4. Finally, in Section 5, we conclude the work along with stating the possible future extensions of the work.

2. Related works

In this section, we give a brief survey of WSD approaches proposed in the literature. First, Lesk-based approaches are discussed. Then, we discuss a few of the graph-based approaches proposed in the literature. Later, we discuss ConceptNet-based approaches. We conclude this section by describing the identified research gap and the approach proposed to fill this gap.

Various dictionary-based and unsupervised approaches have been proposed in the literature, which try to identify the actual sense of a word. Actual sense is identified by exploiting the information derived from the word's context. The algorithms proposed by Lesk (1986), Banerjee and Pederson (2002), Patwardhan, Banerjee, and Pedersen (2003), Basile, Caputo, and Semeraro (2014) exploited pairwise semantic similarity among an ambiguous word and the words in its context. We discuss these approaches below.

In the Lesk algorithm (Lesk 1986), the definition, also known as gloss, of each sense of a word in a sentence is compared to the glosses of every other word in the sentence. The definition of a word is fetched from a dictionary. A word is assigned the sense for whose gloss shares the largest number of words in common with the glosses of the other words. An adapted version of the Lesk Algorithm was proposed by Banerjee and Pedersen (2002), who used WordNet instead of a dictionary for their algorithm. For comparison, the algorithm includes neighboring glosses and those of words related to the word to be disambiguated. Patwardhan *et al.* (2003) generalized this approach as a method of WSD. Basile *et al.* (2014) have proposed an enhanced form of the Lesk algorithm that is based on the distributional semantic model. The algorithm uses a word similarity function defined on a distributional semantic space to compute the gloss-context overlap. Authors of the enhanced Lesk algorithm adapted BabelNet (Navigli and Ponzetto 2012) as sense inventory.

Our work is in the continuation of the approach, where we identify a correct sense of a word by finding an overlap between words in context and glosses prepared using a knowledge base. All of the previously proposed approaches defined context as a collection of words of a sentence. However, in the proposed algorithm, the context includes both the words of a sentence and implicit words that are meant but not written.

Graph-based approaches have gained researchers' attention in the last few years. This is due to the native ability of graphs to organize contextual information as well as compute relations between word senses. This ability enables researchers to extract structural and semantic properties of the text. A few of the examples in the literature include Navigli and Lapata (2007; 2010), Sinha and Mihalcea (2007), Manion and Sainudiin (2014), Agirre, Lacalle, and Soroa (2014), Moro, Raganato, and Navigli (2014), Chaplot, Bhattacharyya, and Paranjape (2015), and Pina and Johansson (2016). We give a brief description of these approaches below.

Navigli and Lapata (2007; 2010) proposed an unsupervised WSD approach based on various measures of graph connectivity to deduce the relative importance of a node in the graph. The approach uses WordNet as a sense inventory as well as a knowledge base. This approach creates a graph, starting with WordNet senses as nodes, and then includes all the nodes and edges that exist on the paths between each pair of senses. It incorporates various local centrality measures and global connectivity measures to check the node's connectivity to all other nodes of the graph. The approach considered local centrality measures viz. degree, closeness, betweenness centrality, and their variants. Global connectivity measures used in the approach are compactness, graph entropy, etc. This approach showed that the local connectivity measures performed better than the global connectivity measures, particularly the degree centrality and PageRank.

Sinha and Mihalcea (2007) created a labeled graph such that there is a vertex for every possible sense for every word of the sentence. A weighted graph of label dependencies is built by adding

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

a vertex for each admissible label and an edge for each pair of labels for which a dependency is identified. A maximum allowable distance can be set (MaxDist), indicating a constraint over the distance between words for which a label dependency is sought. Label dependencies are determined through the dependency function which encodes the relation between word senses. Scores are then assigned to vertices using a graph-based centrality algorithm. Sinha and Mihalcea (2007) experimented with various centrality measures viz. indegree, Closeness, Betweenness, and PageRank. The most likely set of labels is determined by identifying, for each word, the label that has the highest score.

Manion and Sainudiin (2014) introduced an approach named SUDOKU that tries to improve centrality. It is an iterative approach that simultaneously constructs a graph and disambiguates the words using a centrality function.

Agirre *et al.* (2014) proposed a new approach based on random walks, which uses WordNet and extended WordNet to create a graph. The modified PageRank random walk algorithm is used to include the context of the sentence for disambiguation. Acknowledging the intertwining of entity linking and WSD, Moro *et al.* (2014) proposed a unified graph-based approach for entity linking and WSD, calling it Bablefy.

Chaplot *et al.* (2015) proposed a new approach for unsupervised WSD. This approach was based on theories of sense dependency and selective dependency. *Sense dependency* states that the sense of a word does not depend on other words of the sentence but rather depends on the senses of the other words in the sentence. *Selective dependency* states that the sense of an ambiguous word is guided by only a few words of the sentence and their respective senses, and not by all words of the sentence. The goal of this approach is to maximize the joint probability of senses of all the words in a sentence, given a dependency structure of the sentence, frequency of senses, and similarity among them.

Pina and Johansson (2016) proposed another graph-based approach for WSD where sense and context embedding are constructed by applying the Skip-gram method to random walks over the sense graph. Their approach assigns a score to all the senses of an ambiguous word based on the dot product of the sense vector with the sum of the context vectors. It does not use any a priori sense probabilities and instead generates all information from the used knowledge base in the form of co-occurrence of concepts. The word is disambiguated by maximizing the sense score and selecting the sense with the highest score. It is successful in generating faster results compared to other methods.

We see much less work in the literature that uses ConceptNet for WSD. Two of the approaches based on ConceptNet are Blending (Havasi, Speer, and Pustejovsky (2010)) and ExtGM (Chen and Liu 2011). Blending (Havasi *et al.* 2010) uses a common sense reasoning technique, called blending, which combines knowledge from SemCor, WordNet, ConceptNet, Extended WordNet, and their ambiguated versions. The approach also uses the Brown1 and Brown2 corpus of SemCor 3.0 to gain knowledge about words that occur together in a sentence. With blending, a single vector space is created to model semantic similarity and associations, and data sets are included in the matrix using appropriate weighing factors. For disambiguation, an ad hoc category vector is created for each sentence, which represents words and meanings that are likely to appear in the sentence. For each word that is to be disambiguated, the sense of the word with the highest dot product with the ad hoc category's vector is declared as the correct sense of the word.

ExtGM (Chen and Liu 2011) first disambiguates concepts in ConceptNet, then tags each concept in a given assertion with one of the senses. It then combines ConceptNet and WordNet to create an enriched knowledge base. Chen and Liu (2011) did not develop any methodology for WSD, but instead implemented a WSD approach proposed by Galley and McKeown (2003) on their extended knowledge base.

We have identified a research gap while performing the literature survey. We identified that all of the approaches define the context of a sentence as consisting of words of the sentence. None of the approaches consider that the context of a sentence also includes implicit words, which truly

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

guide the meaning of ambiguous words in a sentence. We also identified a high potential in ConceptNet to be used as a knowledge base. The common sense information that it has collected from volunteers makes it a rich and distinguished source of knowledge. In the proposed algorithm, we fetch the implicit words of a sentence using ConceptNet and then perform WSD of the ambiguous word using explicit and implicit information. As shown in the following section, the proposed algorithm successfully detects the implicit words that correctly describe the context of a sentence. The proposed algorithm ensures that a consistent solution of the WSD problem is found.

3. Proposed algorithm for disambiguation

The proposed algorithm proceeds incrementally and works sentence by sentence. Every sentence is processed using the Java-based Apache OpenNLP[c] toolkit. We performed stemming, part-of-speech (POS) tagging and chunking of each sentence using OpenNLP.

This algorithm has four phases:

1. Generate a graph representation of an input sentence.
2. Create an input sentence profile (ISP) of the input sentence.
3. Create a word sense profile (WSP) for every sense of the ambiguous word.
4. Interpret the ambiguous word.

3.1 Generate graph representation of an input sentence

We generate a graph representation of an input sentence to expose the connection between the words of a sentence. An input sentence contains an ambiguous word and other content words. We refer to content words, except for the ambiguous word, as well as phrases of the input sentence as *constituent terms*. The graph of a sentence is generated using constituent terms and ConceptNet. We add all the content words and phrases (NP, VP, ADJP, and ADVP) of the sentence in the initial sentence graph. However, preposition phrases (PPs) are avoided. PP generally has a preposition followed by a noun phrase. Since prepositions are noncontent words, they are not included in the initial graph; and the following noun phrase (NP) is already included. For example, consider the sentence "I play bass in a jazz band." The parse of the given sentence using OpenNLP is *"(TOP (S (NP (PRP I)) (VP (VBZ play) (NP (NN bass)) (PP (IN in) (NP (DT a) (NN jazz) (NN band))))).*" The content words in the sentence are—*play (VBZ), bass (NN), jazz (NN), band (NN)*. The phrases in the sentence are—noun phrase (NP)—*I, bass* and *a jazz band*; verb phrase (VP)—*play, play bass in a jazz band*; and preposition phrase (PP)—*in a jazz band*. Since we only consider content words, *I, a,* and *in* are omitted. Thus, for the given sentence, the constituent terms are *play, bass, jazz, band* and *jazz band*, which are also available in ConceptNet.

The graph is generated using ConceptNet as a knowledge base. We fetch a subgraph of ConceptNet with the ambiguous word as the centre of the graph. The constituent terms, which are spread across the graph, are connected to the ambiguous word via different paths of varying lengths. In ConceptNet, concepts are related to one another via different *Relations*, which provide vital information in understanding common sense knowledge. However, we are only interested in the concepts that are related to one another and not the type of relations that connect the concepts.

The following subsection gives more details on the method of graph creation.

3.1.1 Graph creation using spreading activation

'Spreading Activation is a method for searching associative networks, neural networks or semantic networks' (Collins and Loftus 1975). Experiments performed by various researchers have showcased its efficacy as a methodology for searching graphs (Meyer and Schvaneveldt 1971).

[c] https://opennlp.apache.org/. OpenNLP provides us with predefined models for stemming, POS tagging and chunking of a sentence. OpenNLP implements Porter Stemming algorithm for stemming and OpenNLP's parser is trained on Conll2000 shared task data.

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

We implement spreading activation on ConceptNet to obtain a graph cocooned around an ambiguous word. We initiate the spreading process by assigning an initial weight T to an ambiguous word (w_a). The ambiguous word is the source node from where the activation begins. We used $T = 10$ in our experiments. Then, we spread this activation outward to other nodes linked to the source node. The weight decays with a decay factor (d) as activation propagates throughout the network. We used $d = 1$ in our experiments. The activation values range between 0 and 10. The weight starts at 10 and decays with every activation with a decay factor of $d = 1$. The spreading process stops if there is no node to traverse or if the weight has been reduced to 0. Different values of d affect the decaying value of spreading. If we increase the value of d, then the graph shrinks, resulting in a smaller graph.

Algorithm 1 Algorithm for Spreading Activation

```
Input: ConceptNet graph in the form of adjacency list, $T$ is the weight, $d$ is the decay factor, and
  w is the node to be spread outside. Initially, w=$w_a$, d=1, and T=10.
Output: Graph G, a subset of ConceptNet in the form of a list of edges
  procedure SPREADING_ACTIVATION($w$, $d$, $t$)
      processed[w] ← 1
      for vertex $w_m$ adjacent to $w$ do
          if (w, $w_m$) or ($w_m$, w) is not in G then
              Add Edge (w, $w_m$) to G
              if proceseed[$w_m$] = 0 or T-d $\geq$ 0 then
                  return SPREADING_ACTIVATION($w_m$, $d$, $T - d$)
              end if
          end if
      end for
  end procedure
```

3.2 Create input sentence profile

The ISP of a sentence is a set of words or phrases that help a program understand the relationship between the constituent terms and the ambiguous word of a sentence based on common sense knowledge. We traverse the graph of the input sentence generated in the first step and find paths connecting every constituent term to the ambiguous word to create a profile of a sentence. We then use the depth first search to find the paths. The search starts from the ambiguous word (w_a), and we mark all the constituent terms (W_c) in the graph. Next, we find all paths from these constituent terms to w_a. Every node that lies along these paths becomes a member of the set of supporting terms (W_s) of the ISP. Since the operation is computationally expensive because of the high branching factor of the nodes, finding all the paths of different lengths is difficult and irrelevant. Therefore, we use a Threshold (L) to restrict the path lengths, that is $|pathLength| \leq L$. After performing multiple experiments (refer to Section 4), we fixed $L = 7$.

Let S be a set of constituent terms and ambiguous word of the input sentence. The ISP of a sentence is denoted by *ISP(S)*, such that, ISP(S) = $W_a \cup W_c \cup W_s$, where

W_a = {set of ambiguous words} = {w_a}
W_c = {set of constituent terms of a sentence} = S-{w_a}, includes all the content words and phrases (NP, VP, ADJP, and ADVP) of a sentence, except the ambiguous word.
W_s = {set of supporting terms}, are terms (both words and phrases) that enable a program to deduce the connection between an ambiguous word and constituent terms of the sentence under consideration. All the terms that lie along the paths from constituent terms to the ambiguous word are included in this set of supporting terms.

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

3.3 Create Word Sense Profile (WSP)

We disambiguate ambiguous words by tagging them with a sense using WordNet as a word sense inventory. WordNet is an online lexical database that arranges words in terms of synonyms called synsets. Each synset represents a different sense, and each sense has a definition and a list of synonym words. Synsets are related to one another via lexical or semantic relations such as hypernymy, hyponymy, meronymy, antonymy. A gloss demonstrating sense usage also supports each sense.

This phase consists of the creation of a WSP (Chen and Liu 2011) for all the senses of the ambiguous word under consideration. A WSP is a set of words created using WordNet, and the WSP of a sense includes all synonyms in the synset, content words in their glosses, and words in semantically related synsets. For example, consider the second sense of the word *glass* (noun), which means a container used for drinking. It is represented as $\{glass_n^2\}$ where the subscript denotes the part-of-speech and the superscript denotes a sense number in WordNet. The WSP of a sense of *glass* is a union of all the words in the sets defined below.

WSP($glass_2^n$) = {drinking glass, container, made, glass, holding, liquids, drinking, beer glass, bumper, goblet, . . ., water glass, wine glass, glass, container}
Synonyms in synset: {drinking glass}
Content words in gloss: {container, made, glass, holding, liquids, drinking}
Semantically related synsets include:
Hyponyms: {beer glass, bumper, goblet, . . ., water glass, wine glass}
Meronym: {glass}
Hypernym: {container}

3.4 Interpret ambiguous word

To suggest a sense for the ambiguous word, we separately implement a set intersection operation between every WSP and ISP. In set theory, an intersection of two sets *A* and *B* is a set that contains all elements of *A* that also belongs to *B*. In the proposed algorithm, the ISP of the sentence, as well as all WSPs, are sets of words. On the application of set intersection, we obtain a set of words that belong to both the ISP and the corresponding WSP. The sense with an intersection set of highest cardinality is suggested as the correct sense. A conflict may arise when intersection sets of two or more senses have the highest and equal cardinality. Two such cases may occur. First, these senses are closely related. Second, the senses are different in context. The second case is considered as a failure case, as we cannot conclude a single sense for the ambiguous word. However, further processing is performed in the first case. The first case happens because WordNet is a fine-grained sense inventory where senses are very closely related. In this scenario, we further process these intersection sets, removing those concepts from intersection sets that are directly related to the ambiguous word but remotely to other constituent concepts.

For example, consider the usage of the word *bat* in the sentence “He batted the ball to the boundary.” The ISP of the sentence is {ball, bat, boundary, hitting, smash, bang, beat, club, baseball, cricket, baseball stick, attack, hit six, run, four, score, batting, game, soft ball, stick, wooden stick, sport, racket, tennis, badminton, sports equipment}. There are five senses of the word *bat* in the WordNet. We thus applied intersection operation between the ISP and the WSPs. The highest cardinality intersection sets are $Intersect_4$ = {bat, club, cricket, hitting, sport} and $Intersect_5$ = {bat, club, hitting, ball, stick}. These sets are obtained by the intersection of WSP_4 with the ISP and WSP_5 with the ISP. The cardinalities of both intersection sets are equal. To resolve this, we find the words or phrases that are directly connected to *bat* and are not directly connected to the other constituent terms. We found that from $Intersect_4$, *club* is the only word that is directly connected to *bat* and remotely to others. However, in $Intersect_5$, *club, ball*, and *stick* are directly connected to *bat*. Therefore, we removed {club} from $Intersect_4$ and {club, ball, stick} from $Intersect_5$. After deleting these terms, the updated Intersect sets are given as $Intersect_4$ = {cricket, hitting, sport} and

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

$Intersect_5$ = {hitting}. $Intersect_4$ has a higher cardinality; thus bat_n^4 is deduced as the right sense for the given ambiguous word. This kind of situation occurs in the case of very close senses which are even difficult for a human annotator to judge.

Algorithm 2 Algorithm for WSD using Implicit Information

```
Input: $S$ - a set of constituent terms extracted from the given sentence, $T$ - Threshold value for
    graph creation, $d$ - decay factor, and $L$ - Threshold for path length.

    Let S= {$w_1$, $w_2$, . . ., $w_i$, . . . $w_n$}, where $w_i$ represents the $i^{th}$ $(1 \leq i \leq n)$ word or phrase in the
    input sentence.
    Let $w_a$ be an ambiguous word and constituent terms, $W_c$ = S-{$W_a$}

    Step 1: Create the ISP of the sentence using the following steps:
        (i) Call Spreading_Activation($w_a$, d, T)
        (ii) Locate constituent terms, $W_c$, in the graph and find all paths connecting $w_a$ to these
    concepts such that $|path\_length| \leq$ L.
        (iii) ISP = {$c_i$ | concepts that lie along the paths from $W_c$ to $w_a$ }

    Step 2: Create WSPs of all the senses of the ambiguous word having the same part-of-speech.
    WSP($sense_j$) with $1 \leq j \leq m$, where $m$ is the number of senses that a term has in WordNet.

    Step 3: $Intersect_j$ = WSP($sense_j$) $\cap$ ISP

    Step 4: If only one set, $Intersect_h$, has the highest cardinality, suggest $sense_h$ as the correct sense.
    Else go to Step 5.

    Step 5: If more than one intersection set exists, let $Intersect_x$ and $Intersect_y$, such that their
    cardinalities are maximum and equal. Remove the words from intersection sets ($Intersect_x$ and
    $Intersect_y$) that are directly related to the ambiguous word $w_a$ but remotely to other constituent
    terms, $W_c$. The updated Intersect set with the highest cardinality is given as the correct sense.
```

3.5 Illustrative examples

In this section, we discuss various examples to demonstrate how our proposed algorithm successfully interprets the correct sense of an ambiguous word. Even a graph generated for a simple sentence becomes a complex graph due to the size of ConceptNet. Therefore, for the sake of brevity and to make our examples understandable, we give excerpts of our original graphs, which include only a few of the most relevant concepts. We demonstrate the algorithm for the examples mentioned in Section 1. Since all the example sentences have the same ambiguous word, we define WSPs for all the senses of word $bass_n$ once.

WSP($bass_n^1$) = {bass, lowest part, treble, pitch, concert pitch, soprano, tenor, low frequency, alto, tone, tune, high pitch, key, low pitch, high frequency, musical range};
WSP($bass_n^2$) = {bass, thorough bass, basso continuo, bass part, figured bass, part, voice, secondo, continuo, primo, ground bass, voice part};
WSP ($bass_n^3$) = {bass, vocalizer, musician, player, lowest voice, singer, adult male singer, performer, performing artist, vocalist, entertainer, basso, instrumentalist};
WSP($bass_n^4$) = {bass, sea bass, striper, seafood, food, family Serranidae, solid food, lean flesh, striped bass, saltwater fish, edible fish, shellfish, roe, Atlantic Coast, United States, elongated body, long spiny dorsal fin};

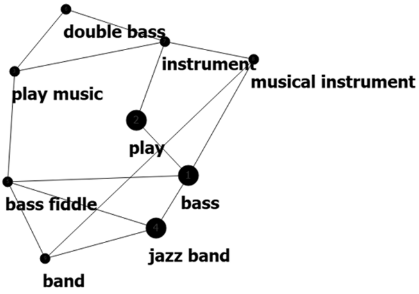

Figure 3. Excerpt of the graphical expansion for the sentence ‘I play bass in a jazz band.’

WSP($bass_n^5$) = {bass, freshwater bass, North American, freshwater fish, lean flesh, flesh, largemouth bass, smallmouth bass, North American food, game fish};
WSP($bass_n^6$) = {bass, basso, basso profundo, very deep bass voice, singing voice, lowest adult male singing voice, singing voice, lowest, voice, vocalization, vocalisation, vocalism, phonation, vox, bass voice, baritone, baritone voice};
WSP($bass_n^7$) = {bass, member, lowest range, musical instrument, bass violin, bull fiddle, double bass, contrabass, family of musical instruments, bass fiddle, string bass, bass guitar, instrument, bass horn, sousaphone, tuba, bombardon, bombard, produce musical tones, sounds};
WSP($bass_n^8$) = {bass, nontechnical name, percoid, edible marine, percoid fish, percoidean, freshwater spiny-finned fish, freshwater bass}.

(i) I play *bass* in a jazz band.

An excerpt of graphical expansion for this sentence is shown in Figure 3. We can see that the constituent terms, *jazz band* and *play*, are directly related to the ambiguous word (*bass*). Obtaining constituents terms early in the graph makes the task more comfortable, but this does not help provide any information about how these terms are related to one another via common sense knowledge. This also demonstrates that a small value of the Threshold (L) is not a very good choice to opt for.

For any given sentence, the ISP is represented as ISP(S) = $W_a \cup W_c \cup W_s$. For the given sentence, W_a = {bass}; W_c = {play, jazz band}; and W_s = {double bass, play music, bass fiddle, instrument, musical instrument, band}. Therefore, the ISP for the given sentence is ISP(S) = {bass, play, jazz band, double bass, play music, bass fiddle, instrument, musical instrument, band}. Next, we find the intersection of ISP(S) with all the WSPs defined above. $Intersect_1$ = {bass}; $Intersect_2$ = {bass}; $Intersect_3$ = {bass}; $Intersect_4$ = {bass}; $Intersect_5$ = {bass}; $Intersect_6$ = {bass}; $Intersect_7$ = {bass, musical instrument, instrument, bass fiddle, double bass}; $Intersect_8$ = {bass}. Looking at the Intersect sets, we can conclude that $Intersect_7$ has the highest cardinality and therefore $bass_n^7$ is the deduced right sense for the ambiguous word.

(ii) He was hungry and cooked *bass*.

Following a similar approach used for the creation of the ISP for the previous sentence, the ISP created for this sentence is {hungry, cook, bass, eat, food, hunger, fish, cook food, seafood, freshwater fish, freshwater bass, eat food} (refer Figure 4). The intersection sets generated as a result of intersection operation between the ISP and WSPs are $Intersect_1$ = {bass}; $Intersect_2$ = {bass}; $Intersect_3$ = {bass}; $Intersect_4$ = {bass, food, seafood}; $Intersect_5$ = {bass, freshwater fish, freshwater bass}; $Intersect_6$ = {bass}; $Intersect_7$ = {bass}; $Intersect_8$ = {bass, fish, freshwater bass}. The intersection sets with the highest cardinality are $Intersect_4$, $Intersect_5$, and $Intersect_8$. Now, by applying Step 5 of our algorithm, no removals occur from $Intersect_4$. Both *freshwater fish* and *freshwater bass* are removed from $Intersect_5$, as they are directly related to *bass* but remotely related to *cook* and *hungry*. From $Intersect_8$, only *freshwater bass* is deleted for the reason mentioned above. However, *fish* remains within $Intersect_8$, as this is directly related with *hungry*. Therefore, the

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

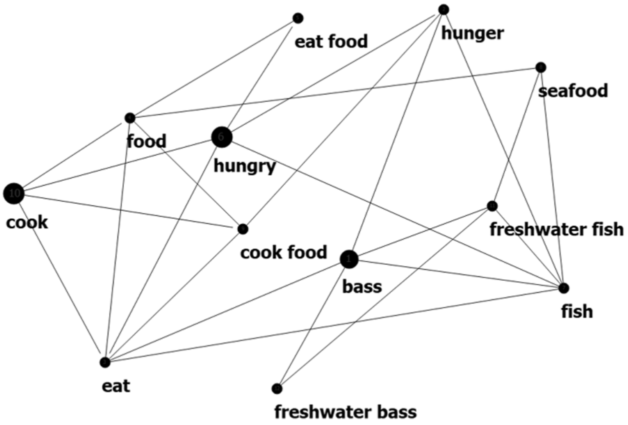

Figure 4. Excerpt of the graphical expansion for the sentence ‘He was hungry and cooked bass.’

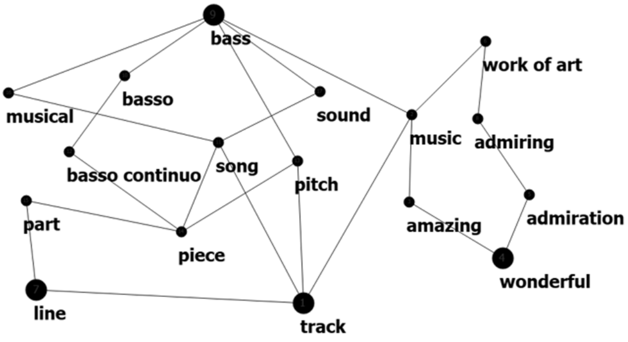

Figure 5. Excerpt of the graphical expansion for the sentence ‘The track has wonderful bass line.’

updated intersection sets are *Intersect*$_4$ = {bass, food, seafood}; *Intersect*$_5$ = {bass}; *Intersect*$_8$ = {bass, fish}. *Intersect*$_4$ is of highest cardinality. Therefore, the deduced sense for *bass* in the given context is $bass_n^4$.

(iii) The track has a wonderful *bass* line.

The graph (Figure 5) shows the connectivity between *track* and *bass* via different paths. One path is *track-music-bass* and another path is *track-song-piece-basso continuo-basso-bass*. A music listener relates with the first path, whereas a music composer relates closely with the second path. Thus, the graph uncovers the relationship between terms based on the knowledge of different people. The ISP for the given sentence is ISP = {bass, wonderful, track, line, pitch, basso, basso continuo, amazing, admiration, admiring, work of art, music, part, piece, song, sound} (refer Figure 5). The intersection of the ISP with the WSPs generates the following intersect sets. *Intersect*$_1$ = {bass, pitch}; *Intersect*$_2$ = {bass, basso continuo, part}; *Intersect*$_3$ = {bass}; *Intersect*$_4$ = {bass}; *Intersect*$_5$ = {bass}; *Intersect*$_6$ = {bass, basso}; *Intersect*$_7$ = {bass}; *Intersect*$_8$ = {bass}. Among these obtained intersect sets, *Intersect*$_2$ has the highest cardinality. Therefore, $bass_n^2$ is the correct sense of the word *bass* in the sentence.

(iv) Please install the *patch*.

In the given input sentence, *patch* is the ambiguous word. The WSPs for all the senses of the word *patch* as a noun are listed below.

WSP($patch_n^1$) = {patch, spot, speckle, dapple, fleck, maculation, marking, design, pattern, figure, crisscross, cross, mark};

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

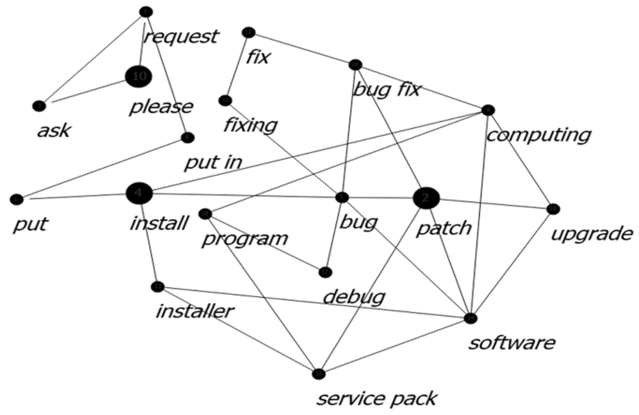

Figure 6. Excerpt of the graphical expansion for the sentence 'Please install the patch.'

WSP($patch_n^2$) = {patch, plot, plot of land, plot of ground, small area. Ground, vegetation, bean plot, cabbage patch, tract, piece of land, piece of ground, parcel of land, parcel};
WSP($patch_n^3$) = {patch, piece of cloth, piece of material, decoration, mend, cover, fabric, piece, part, portion};
WSP($patch_n^4$) = {patch, while, piece, spell, period, indeterminate length, bad weather};
WSP($patch_n^5$) = {patch, commands, correct, bug, computer program, programme, computer programme, instructions, computer, execute, interpret, software, package, code};
WSP($patch_n^6$) = {patch, temporary hookup, connection, limited time, connection, connecter, connector, connective, connexion, lash-up, contrivance};
WSP($patch_n^7$) = {patch, mend, darn, sewing, repair, worn, torn hole, stitchery, needlework, needle, thread, needlecraft, binding, stitch, applique, binding, patchwork};
WSP($patch_n^8$) = {patch, eyepatch, cloth, covering, eye, cloth covering, artifact, artefact, covers, protect, shelter, conceal};
WSP($patch_n^9$) = {patch, bandage, injured, part of body, covers, protects, adhesive bandage, capeline bandage, cast, plaster cast, compression bandage, amputation, medical, dressing, gauze, plastic, fabric tape, adhesive}.

A part of graphical expansion for the sentence is given in Figure 6. The ISP for the sentence is ISP ={patch, software, bug, computing, program, installer, service pack, upgrade, debug, please, request, put in, put, fix, bug fix}. The intersection of the ISP and WSPs generates the following intersection sets. $Intersect_1$ = {patch}; $Intersect_2$ = {patch}; $Intersect_3$ = {patch}; $Intersect_4$ = {patch}; $Intersect_5$ = {patch, software, bug}; $Intersect_6$ = {patch}; $Intersect_7$ = {patch}; $Intersect_8$ = {patch}; $Intersect_9$ = {patch}. As is evident, the intersection set $Intersect_5$ has the highest cardinality. Therefore, $patch_n^5$ is the correct sense for the word *patch* in the given sentence.

To demonstrate the effectiveness of the proposed algorithm on a verb, such as "be", let us take an example sentence of "He is a good singer." The word to be disambiguated is *is*. WordNet indicates "be" as the base form of the verb "is", and thus we use *be* as the word to be disambiguated. The ISP of the sentence is {be, good, singer, to be, being, am, music, is, human, human being, person, artist, music}. Then we create the WSPs for all 13 senses of the word *be* in WordNet. The WSPs are given below:

WSP(be_v^1) = {be, is, equality, being, John, rich, good, answer, look, suffer, hurt, be well, feel, stay, remain, rest, continue, sparkle, scintillate, coruscate, confuse, throw, fuddle, befuddle, confound, rank, point, want, need, require, compact, pack, rest, cut};
WSP(be_v^2) = {be, is, identical, someone, something, president, John Smith, house};

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

WSP(be_v^3) = {be, occupy, certain position, area, somewhere, where, umbrella, toolshed, is, back, what, behind, behavior, this, stretch, stretch along, attend, go to, fill, populate, dwell, live, inhabit, reach, extend to, touch, run, go, pass, lead, extend};
WSP(be_v^4) = {be, exist, existence, extant, God, there, come, preexist, coexist, cohabit, indwell, prevail, hold, obtain, consist, lie in, lie, distribute, dwell};
WSP(be_v^5) = {be, happen, occur, take place, lost, wallet, this, during, visit, parent's house, there, two hundred people, funeral, lot, noise, kitchen};
WSP(be_v^6) = {be, equal, identical, equivalent, one dollar, equals, 1000 rubles, days, equate, correspond, match, fit, check, jibe, gibe, tally, agree, represent, stand for, correspond, translate};
WSP(be_v^7) = {be, constitute, represent, make up, comprise, form, compose, make, form, range, straddle, fall into, fall under, present, pose, supplement, money, income, stone wall, backdrop, performance, constitute, entire, belonging, children, made up, chorus, sum, represents, income, year, few, men, army};
WSP(be_v^8) = {be, follow, work, specific, place, specific place, specific subject, function, herpetologist, resident philosopher, vet, cox};
WSP(be_v^9) = {be, embody, personify, represent, character, stage, Derek Jacobi, Hamlet, body, exemplify, typify, symbolize, symbolize, stand for};
WSP(be_v^{10}) = {be, spend, use time, hour, take, occupy, use up, space};
WSP(be_v^{11}) = {be, live, have life, alive, great leader, no more, grandfather, lived, end of war, survive, last, live on, go, endure, hold up, hold out, living};
WSP(be_v^{12}) = {be, remain, unmolested, undisturbed, uninterrupted, infinitive form, let, stay, rest};
WSP(be_v^{13}) = {cost, be, priced at, set back, knock back, put back}.

Next, we perform an intersection of the ISP with the WSPs. We determine that $Intersection_1$ = {being, is, good} has the highest cardinality and thus be_n^1 is deduced as the correct sense of *be* in the above example.

4. Experimental results and discussion

In this section, we first describe how the Threshold value (L) for the proposed algorithm has been found. Next, we describe various performed experiments and obtained results and their interpretation. To evaluate the performance of our algorithm, we compute three measures, viz. precision, recall, and f1-measure. Since our algorithm provides an answer to all ambiguous words, the values for precision, recall, and f1-measure are the same.

- *Precision (P)*—the percentage of correct senses out of the total identified senses;
- *Recall (R)*—the percentage of the number of correctly identified senses to the total number of words in the data set; and
- *F1-measure*—combination of precision and recall, F1-measure = (2*P*R/ (P+R)).

4.1 Tuning of threshold (L) value

For the proposed algorithm, the value of Threshold (L) is tuned using two data sets viz. SemEval-2010 task 17 (Agirre *et al.* 2010) and SemEval-2015 task 13 (Moro and Navigli 2015). The text in the first data set is from the ecology domain, whereas the text in the second data set is composed of data from four different domains: medical, drug, math, and social issues. A total of 794 words out of 1,398 words were disambiguated in the first data set, and the highest F1-measure of 56.79 was obtained at $L = 7$. A total of 819 words were disambiguated out of 1,261 words from the second test set and the highest F1-measure of 64.96 was obtained at $L = 7$. The results are shown in Tables 1 and 2. With this observation, we set the Threshold (L) for our algorithm at $L = 7$.

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

Table 1. Impact of Threshold (L) on the F1-measure of the proposed algorithm for the SemEval-2010 Task 7 data set

Threshold (L)	F1-measure
5	51·21
6	55·87
7	56·79
8	54·30
9	49·55

Table 2. Impact of Threshold (L) on the F1-measure of the proposed algorithm for the SemEval-2015 Task 13 data set

Threshold (L)	F1-measure
5	61·23
6	63·88
7	64·96
8	63·62
9	60·33

4.2 Experiments performed

To test our algorithm, we ran it with two distinct data sets. One was provided for *SemEval-2013 Task 12: Multilingual Word Sense Disambiguation* and other for *SemEval 2007 Task 7: Coarse-Grained English All-Words Task*. One of the significant differences in these two data sets is their sense granularities. While the SemEval-2007 task uses coarse-grained word senses, the SemEval-2013 task relies on fine-grained distinctions. Since only nouns are tagged in the SemEval-2013 task, we also tested our algorithm with the SemEval-2007 test data set, which is an all-word WSD task. A scorer was provided along with the data sets to ensure consistent scoring of the algorithms. We also obtained our results using the official scorer of these tasks.

4.2.1 Testing on SemEval-2013 Task 12: Multilingual word sense disambiguation

We ran our algorithm on a test set of Semeval-2013 task for multilingual word sense disambiguation (Navigli, Jurgens, and Vannelle 2013). The test set consisted of 13 articles covering different domains. For English, there were 1644 words of running text consisting of 1502 single-words and 85 multiword expressions. For the task, participating systems were given the flexibility to use either BabelNet or its two main subsets, viz. WordNet and Wikipedia, as a sense inventory. We used WordNet as a sense inventory for annotating ambiguous words in our experiments. To do so, we restricted ourselves to items annotated with BabelNet synsets that contain WordNet senses for the ambiguous word.

The proposed algorithm correctly tagged 1012 words out of 1502 single-words and all the multiword expressions when tested with the test set. It did so with an F1-measure of 69.13 at $L = 7$. However, the highest F1-measure of 69.30 was achieved with $L = 6$. Table 3 has a comparison of the results of the proposed algorithm with the results of the algorithms that were submitted in the SemEval-2013 task. We have also compared our algorithm with the most frequent sense

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

Table 3. Comparison results with systems participating in SEMEVAL-2013 Task 12

Team	System	F1-measure
GETALP	WN-1	51·4
UMCC-DLSI	RUN2	64·7
Proposed algorithm		69·13
MFS		63·0

Table 4. F1 scores for 13 articles in the SemEval-2013 Task 12 (for WordNet)(T.=Translated)

Text	Description[a]	Length	F1-measure
1	General environment	194	70·2
2	T. Politics	76	52·7
3	T. Economics (Wall street)	69	62·1
4	News, General	109	66·9
5	T. Economics (Banks)	53	49·2
6	Web general	190	69·8
7	T. Sports	178	45·2
8	Science	135	65·1
9	Geopolitics economics	159	61·4
10	General law	138	64·9
11	T. Sport	76	59·7
12	T. Political	170	59·1
13	T. Economics (Deutsche bank)	97	63·2

[a] (Schwab *et al.* 2013)

Table 5. Impact of Threshold (L) on the F1-measure of the proposed algorithm for the SemEval-2013 test set

Threshold(L)	F1-measure	Threshold(L)	F1-measure
1	35·64	6	69·30
2	42·90	7	69·13
3	49·97	8	67·29
4	58·73	9	63·71
5	65·41	10	62·90

(MFS) baseline. MFS of an ambiguous word is the sense with maximum frequency in the training data. Table 4 depicts the document level F1 scores for the data set, and Table 5 depicts the influence of *L* on the results for the 2013 data set.

We also performed an investigation with Wikipedia and BabelNet sense inventories. Wikipedia sense synset consists of a page title with an optional label. The structured relations include redirect pages, disambiguation pages, internal links, interlanguage links, and categories. For example,

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

Table 6. Impact of Threshold (L) on the F1-measure of the proposed algorithm for the SemEval-2007 Task 7 data set

Threshold(L)	F1-measure	Threshold(L)	F1-measure
1	38·91	6	69·94
2	44·25	7	79·63
3	54·30	8	78·47
4	59·22	9	77·36
5	68·73	10	72·68

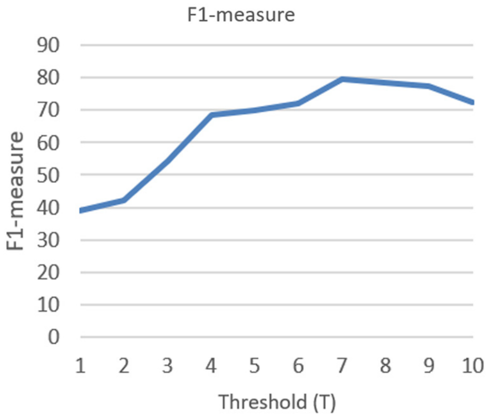

Figure 7. Graphical representation of Threshold (L) versus F1-measure.

consider the word "play". Its synset consists of {play, play(theatre)}. With this little information in a synset, the results were not very encouraging with our algorithm, but the results we obtained from BabelNet are quite inspiring. However, the purpose of our research was to investigate the applicability of WordNet because it is available for Indian languages too, whereas BabelNet does not incorporate most Indian languages.

4.2.2 Testing on SemEval-2007 Task 7: Coarse-grained english all-words task

The SemEval 2007 test set for coarse WSD (Navigli and Lapata 2007) contains five documents belonging to different domains, viz. journalism, book review, travel, computer science, and biography. Out of 2269 ambiguous words, 1807 words were tagged correctly using a Threshold (L) value of 7. This gives an F1-measure of 79.63 (1807/2269).

In Table 6, we summarize the impact of the Threshold (L) on the proposed algorithm. Performance of the algorithm improved with an initial increase in the value of L. However, after a certain limit, the performance began to deteriorate with an increase in the value of L. The relationship between L and F1-measure is shown in Figure 7. The reason for such behaviour is that as the graph size increased with an increasing L, more concepts came into consideration. These concepts were from diverse domains and were not related to our domain of discourse. This hampers the performance of the algorithm.

Out of 2269 word instances, 678 words were monosemous and 1591 were polysemous per the coarse sense inventory. The proposed algorithm correctly tagged all of the monosemous words and 1159 of the polysemous words. Table 7 below summarizes these statistics.

During our experiments, we noticed that a few senses were incorrectly identified because of the unavailability of information in ConceptNet. The reason for this is that common sense knowledge is vast, and ConceptNet is yet not an exhaustive resource that has captured all of the

Table 7. Statistics of the performance of the proposed algorithm for monosomy and polysemy

	Monosemous		Polysemous	
	Instances	Correct	Instances	Correct
Noun	358	358	750	594
Verb	86	86	505	382
Adjective	141	141	221	136
Adverb	93	93	115	47

The first two numeric columns give the number of instances and the number of correct instances of monosemous words. The last two numeric columns represent the total number of instances and the number of correct instances for polysemous words.

Table 8. Performance of different unsupervised coarse WSD algorithms on the SemEval-2007 test set

Algorithm	F1-measure	Methodology
ExtGM	82·8	Unsupervised
Blending	80·8	Unsupervised
Proposed algorithm	79·63	Unsupervised
SUSSX-FR	77·04	Unsupervised
WFS baseline	63·44	
MFS baseline	78·9	
Random baseline	62·7	

available knowledge. Furthermore, many sentences that are available in ConceptNet are incorrect and meaningless. For example, sentences such as “It is a tree” with assertion It→ ISA → Tree fail to provide any relevant information. Dangling sentences such as “Tree is a often” are incomplete and ambiguous. Such sentences were added into the system because volunteers of the OMCS project created the sentences and there were no quality controls to prevent volunteers from entering meaningless data. The presence of such irrelevant information degrades the performance of the proposed algorithm.

There are only a few algorithms that make use of common sense knowledge for WSD. We compared the performance of the proposed algorithm with the ExtGM approach (Chen and Liu 2011), Blending Approach (Havasi *et al.* 2010), and SUSSX-FR (Koeling and McCarthy 2007). The ExtGM approach (Chen and Liu 2011) disambiguates assertions of ConceptNet and then performs WSD. The Blending Approach (Havasi *et al.* 2010) makes use of ConceptNet. SUSSX-FR (Koeling and McCarthy 2007) is the best-unsupervised WSD system that participated in the Semeval-2007 coarse-grained all-words WSD task. We have taken various baseline approaches to compare the performance of the proposed algorithm viz. WFS (WordNet First Sense) baseline, MFS (Most Frequent Sense) baseline, and RB (Random Baseline). The results of all these algorithms along with the proposed algorithm are shown in Table 8.

The approaches ExtGM (Chen and Liu 2011) and Blending (Havasi *et al.* 2010) gave better results than our proposed algorithm. The approach by Chen and Liu (2011) first disambiguates concepts of ConceptNet using WordNet as a sense inventory. It then merges disambiguated

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

Table 9. Performance of the proposed algorithm on different data sets

		F1-measure	
S. No.	Dataset	(By proposed algorithm)	Threshold(L)
1	SemEval 2013 fine-grained	69·3	6
2	SemEval 2007 coarse-grained	79·63	7
3	SemEval 2007 fine-grained	57·23	7
4	SemEval 3 fine-grained	63·01	7
5	SemEval 2 fine-grained	66·67	6

ConceptNet with WordNet and performs WSD of text using this enriched resource. The blending approach proposed by Havasi *et al.* (2010) blends various knowledge bases viz. ConceptNet, WordNet, Extended WordNet, and their ambiguated versions. They also use the Brown1 and Brown2 corpus of SemCor 3.0 to gain knowledge about words that occur together in a sentence. Thus, their approach is also corpus dependent. The blending approach was implemented solely for coarse WSD. In the proposed algorithm, we neither disambiguate ConceptNet, which itself is an enormous task, nor do we merge varied knowledge bases, which is computationally tedious. Instead, we traverse the ConceptNet graph and include all the concepts lying in the path from the constituent terms to the ambiguous word. This gives us implicit terms that connect the words/terms of the sentence together. We then use this information to disambiguate the words in the text. The proposed algorithm is simpler than both of the proposed approaches, and the results are close to current state-of-art algorithms.

We also tested the proposed algorithm on five more data sets. The results obtained from these tests reinforce our decision to use 7 as the value for the Threshold (L). The results are shown in Table 9. For three data sets, the proposed algorithm gave the best results when L was set to 7. For the other two data sets, the F1-measure was highest when L was set to 6. However, the F1-measure at $L = 6$ was only marginally higher than the F1-measure at $L = 7$. Therefore, taking the above observations into consideration, we proposed $L = 7$ to be used for the test sets.

5. Conclusion and future work

WSD is an essential task in NLP, and context plays an important role in disambiguating an ambiguous word. The context of a sentence is comprised of the words in the sentence as well as implicit information, which is not said aloud. The work presented here successfully demonstrated that the knowledge of implicit information could significantly improve a computer program's power to disambiguate an ambiguous word. The proposed algorithm has an accuracy of 69.3 percent, which is higher than all the unsupervised WordNet-based systems submitted in the SemEval-2013 competition. We were successful in disambiguating 79.63 percent of the lexical sample data set of Semeval-2007.

The proposed algorithm is a very humble attempt at disambiguation, and there is large scope for further improvement. Common sense knowledge is extremely significant, and an enormous amount of knowledge needs to be added in ConceptNet. Therefore, we will use OpenIE approaches in our future work to collect as many assertions as possible, which will increase the ConceptNet knowledge base. We will also attempt to include the relationships between the nodes of ConceptNet in the proposed algorithm. This will help create a better understanding of the context of the sentence, and thus a better WSD. We also plan to extend the proposed algorithm to work with Indian languages, and we will attempt to merge WordNet into ConceptNet for better evaluation.

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

References

Agirre E., Lacalle O.L, Felbaum C., Hsieh S., Tesconi M., Monachini M., Vossen P. and Segers, R. (2010). SemEval-2010 task 17: All-words word Sense Disambiguation on a specific domain. In *Proceedings of the 5th International Workshop on Semantic Evaluations, ACL 2010*, pp. 75–80.

Agirre E., Lacalle O.L and Soroa, A. (2014). Random walks for knowledge based word sense disambiguation. *Computational Linguistics* **40,** 57–84.

Banerjee S. and Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using Wordnet. In: Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2002. *Lecture notes in Computer Science*, vol 2276, pp. 136–145. Springer, Berlin, Heidelberg.

Basile P., Caputo A. and Semeraro G. (2014). An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, 25th International Conference on Computational Linguistics*, pp. 1591–1600.

Carpuat M. and Wu D. (2005). Word sense disambiguation vs. statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, Association for Computational Linguistics, pp. 387–394. Stroudsburg, PA, USA.

Carpuat M. and Wu D. (2007). Improving statistical machine translation using word sense disambiguation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL'07)*, pp. 61–72.

Chan Y.S., Ng H.T. and Chiang D. (2007). Word sense disambiguation improves statistical machine translation. In *Proceedings of Annual Meeting-Association for Computational Linguistics, ACL-2007*, pp. 33–40.

Chaplot D.S., Bhattacharyya P. and Paranjape A. (2015). Unsupervised word sense disambiguation using Markov random field and dependency parser. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Texas, Austin, pp. 2217–2223.

Chen J. and Liu J. (2011). Combining WordNet and ConceptNet for word sense disambiguation. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pp. 686–694.

Collins A.M. and Loftus E.F. (1975). A spreading activation theory of semantic processing. *Psychological Review* **82**, 407–428.

Dunmore D. (1989). Using contextual clues to infer word meaning: an evaluation of current exercise types. *Reading in a Foreign Language* **6**, 337–347.

Fellbaum C. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Fengning Y. (1994). Context clues - a key to vocabulary development. *People's Republic of China-forum* **32**(3), 39.

Galley M. and McKeown K. (2003). Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pp. 1486–88.

Havasi C., Speer R. and Alonso, J. (2007). ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, September, Borovets, Bulgaria, pp. 27–33.

Havasi C., Speer R. and Pustejovsky J. (2010). Coarse word sense disambiguation using common sense. In *Common Sense Knowledge: papers from AAAI Fall Symposium.*

Hoey M. (2005). *Lexical Priming. A New Theory of Words and Language*. London: Routledge.

Hung C. and Chen S.-J. (2016). Word sense disambiguation based sentiment lexicon for sentiment analysis. *Knowledge Based Systems* **110,** 224–232.

Kilgarriff A. (1997). I don't believe in word senses. *Computers and the Humanities* **31**, 91–113.

Koeling R. and McCarthy D. (2007). Sussx. WSD using automatically acquired predominant senses. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 314–317.

Lesk M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC'86: Proceedings of the 5th annual international conference on Systems documentation*, pp. 24–26.

Manion S.L. and Sainudiin R. (2014). An iterative sudoku style approach to subgraph-based word sense disambiguation. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (* SEM 2014)*, pp. 40–50.

McCarthy D., Koeling R., Weeds J. and Carroll J. (2007). Unsupervised acquisition of predominant word senses. *Computational Linguistics* **33**, 553–590.

Meyer D.E. and Schvaneveldt R.W. (1971). Facilitation in recognizing pairs of words: evidence of a dependence between retrieval operations. *Journal of Experimental Psychology* **90**, 227–234.

Mihalcea R. (2005) Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 411–418.

Miller G. (1995). WordNet: a lexical database for english. *Communications of the ACM* **38**, 39–41.

Moro, A. and Navigli, R. (2015). SemEval-2015 task 13: multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015)*, pp. 288–297.

Moro A., Raganato A. and Navigli R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics* **2**, 231–244.

Navigli R. (2009). Word sense disambiguation: a survey. *ACM Computing Surveys* **41**, 10.1–10.69.

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press

Navigli R., Jurgens D. and Vannella D. (2013). SemEval-2013 task 12: multilingual word sense disambiguation. In *Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 222–231.

Navigli R. and Lapata M. (2007). Graph connectivity measures for unsupervised word sense disambiguation. In *Proceedings of IJCAI'07*, pp. 1683–1688.

Navigli R. and Lapata M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**, 678–692.

Navigli R. and Litkowski K.C. (2007). SemEval- 2007: task summary. SemEval Web site. http://nlp.cs.swarthmore.edu/semeval/tasks/task07/summary.shtml.

Navigli R. and Ponzetto S. (2012). BabelNet: the automatic construction, evaluation, and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* **193**, 217–250.

Pace-Sigge M. (2013). The concept of lexical priming in the context of language use. *ICAME Journal* **37**, 149–173.

Panman O. (1982). Homonymy and polysemy. *Lingua* **58**, 105–136.

Patwardhan S., Banerjee S. and Pedersen T. (2003). Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of CICLing'03*, pp. 241–257.

Pina L.N. and Johansson R. (2016). Embedding sense for efficient graph-based word sense disambiguation. In *Proceedings of 2016 Workshop on Graph Based Methods for Natural Language Processing, NAACL-HLT*, pp. 1–5.

Rapp R. and Zock M. (2014). The CogALex-IV shared task on the lexical access problem. In *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon*, pp. 1–14.

Rentoumi V., Giannakopoulos G., Karkaletsis V. and Vouros G.A. (2009). Sentiment analysis of figurative language using a word sense disambiguation approach. In *Proceedings of RANLP*, pp. 370–54d.

Schadd F.C. and Roos N. (2015). Word-sense disambiguation for ontology mapping: concept disambiguation using virtual documents and information retrieval techniques. *Journal on Data Semantics* **4**, 167–186.

Schwab D., Tchechmedjiev A., Goulian J., Nasiruddin M., Sérasset G. and Blanchon H. (2013). GETALP: Propagation of a Lesk Measure through an ant colony algorithm. In *Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 232–240.

Singh P., Lin T., Mueller E.T., Lim G., Perkins T. and Zhu W.L. (2002). Open mind common sense: knowledge acquisition from general public. In *Proceedings of On the Move to Meaningful Internet Systems*, pp. 1223–1237.

Sinha R. and Mihalcea R. (2007). Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of ICSC*, pp. 363–369.

Sumanth, C. and Inkpen, D. (2015). How much does word sense disambiguation help in sentiment analysis of micropost data? In *Proceedings of the 6th (WASSA 2015)*, pp. 115–121.

Tratz S., Sanfilippo A., Gregory M., Chappell A., Posse C. and Whitney P. (2007). PNNL: a supervised maximum entropy approach to word sense disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 264–267.

Tripodi R. and Pelillo M. (2017). A Game theoretic approach to word sense disambiguation. *Computational Linguistics* **43**, 31–70.

Vickrey D., Biewald L., Teyssier M. and Koller D. (2005). Word-sense disambiguation for machine translation. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 771–778.

von Ahn L., Keida M. and Blum M. (2006). Verbosity: A game for collecting common sense facts. In *Proceedings of SIGCHI Conference on Human Fctors in Computing Systems*, pp. 75–78.

Wang T., Rao J. and Hu, Q. (2014). Supervised word sense disambiguation using semantic diffusion kernel. *Engineering Applications of Artificial Intelligence* **27**, 167–174.

Weaver W. (1955). Translation. *Machine Translation of Languages* **14**, 15–23.

Wettler M., Rapp R. and Sedlmeier P. (2005). Free word associations correspond to contiguities between words in texts. *Journal of Quantitative Linguistics* **12**, 111–122.

Zhong Z. and 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. and Ng H.T. (2012). Word Sense Disambiguation improves information retrieval. In *Proceedings of ACL 2012*, pp. 273–282.

Cite this article: Jain G and Lobiyal D.K (2020). Word sense disambiguation using implicit information. *Natural Language Engineering* **26**, 413–432. https://doi.org/10.1017/S1351324919000421

https://doi.org/10.1017/S1351324919000421 Published online by Cambridge University Press