International Journal of Engineering & Technology, 7 (3.20) (2018) 428-434

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Metaheuristic for Word Sense Disambiguation: a Review

Wafaa AL-Saiagh[1], Sabrina Tiun[2], Ahmed AL-Saffar[3], Suryanti Awang[4], A. S. Al-khaleefa[5]

[1,2]Knowledge Technology Research Group (KT), Centre for Artificial Intelligent (CAIT), Universiti Kebangsaan Malaysia UKM Bangi, Selangor, Malaysia
[3,4]Faculty of Computer System and Software Engineering, University Malaysia Pahang UMP, Malaysia.
[5]Broadband and Networking (BBNET) Research Group, Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia
Coresponding Arthur Email: wafaa.alsaiagh@siswa.ukm.edu.my

Abstract

Word Sense Disambiguation (WSD) is the process of determining the exact sense of a particular word in accordance to the context in a computational manner. Such task plays an essential role in multiple fields of study such as Information Retrieval and Information Extraction. With the complexity of human language, WSD came up to solve the problem behind the ambiguity between senses in which a single word would yield different meaning. In this vein, determining the exact meaning of the certain word would facilitate the process of identifying the category of such text, accurate corresponding search results and providing an accurately summarized portion. Several approaches have been proposed for the WSD including statistical, semantic and machine learning techniques. This paper aims to provide a review of such approaches by tackling and categorizing the related works in accordance to the main types.

Keywords: *Word Sense Disambiguation, Machine learning Techniques, Semantic Similarity Measurement-Heuristic, Natural Language Processing.*

1. Introduction

In a given text, the most appropriate senses are assigned to words with the help of word sense disambiguation (WSD) [58]. A predefined sense inventory is employed to assign senses in stand-alone WSD systems. This sense inventory has a collection of predefined senses installed to it that can be applied to different words of a certain language, i.e. a lexicon, a dictionary, or a wordnet. Even though there exist some criticisms and problems, e.g. for sense distinctions at the granularity level, the most popularly employed sense inventory is the Princeton WordNet [50] for English WSD. WordNet’s availability and coverage has made it a popular choice for use as a sense inventory in WSD [13].

Two types of tasks can be distinguished in Natural Language Processing (NLP). The first one is the final tasks which perform for their own such as machine translation, information extraction and automatic summarization. The other type is the intermediate tasks, which perform to aid final tasks, such as part-of-speech tagging, identification of morphological root, parsing, and word sense disambiguation.

As long as WSD is one of the intermediate tasks, so it will be useful for some final tasks such as machine translation and information retrieval.

This paper aims to accommodate a review of WSD task by investigating the techniques that have been proposed in the literature. In addition, this paper will investigate the semantic similarity measures that have been used in the previous studies. Finally, a new trend in the area of WSD will be tackled which is the metaheuristic-based WSD. Next sections will tackle these issues in detail.

2. WSD Approaches

In computational linguistics, right from the 1950s up to recent years, WSD has been an active area of research [31]; [3]; [50]. Most of the work on WSD has been on English language [34]. The lack of appropriate resources, especially in the form of sense-annotated corpus data, has been one such factors affecting WSD research for other languages. WSD systems consider sense-annotated corpora as gold standards for training, evaluation and development. As such, a steady progress in the performance and development of WSD algorithms has not been a surprise for languages such as English, for which there are many large sense-annotated corpora, and considerably less on languages that have lesser availability of such corpora. All machine learning approaches commonly use corpora as knowledge sources; however, they differ in the exact task they perform.

Generally, there are four conventional WSD approaches, i.e., supervised, semi-supervised, knowledge-based, and unsupervised approaches. Also, WSD there are some methods that integrate two approaches to reinforce the process of word disambiguation. Specifically, the unsupervised methods keen to invoke knowledge-based assets to gain more necessary features to facilitate the classification process [6]. Alternatively, knowledge-based methods have been generalized by using unsupervised scheme to search for the suitable sense for bag of words [16].

One of the main advantages of unsupervised machine learning approaches is being independent of sense-annotated corpora as they employ non-annotated corpora to cluster word senses, which make them least affected by the knowledge acquisition bottleneck. However, the downside of this approach is the much difficulty involved in the evaluation of the word sense induction task when compared with the WSD classification task. The main reason behind this difficulty is the lack of clear criteria on judging the quality of word sense clusters [50].

2.1 Supervised Machine Learning Approaches to WSD

Supervised approaches to WSD make use of supervised machine learning methods to correctly assign senses to a word. The task at hand could result in a classification problem, where the class requiring prediction could be the corresponding word sense (from a given sense inventory). Since for each lemma, there is a difference in the sets of word senses as well as of classes to be predicted, for each lemma, classification and training of supervised WSD systems are performed separately [51], [64], [29], [39], and [18]. Classifying each word lemma separately is also referred to as word-experts [12].

2.1.1. Co-Occurrence

Encoding ML features often employ occurrence of words in the target word's context. This could be due to features, which have already proven to perform well, are still easily available and applicable to all words and word classes, i.e. they are popular in the literature [38];[40];[45]; [48].

2.1.2. Parts of Speech

Similarly, information about the parts of speech (POS) for words occurring in that context for the target word is made available to all target words. One commonly used feature is machine learning to encode POS information on the target word itself or on context words [38]; [40]; [45]; [36];[48].

2.1.3 Syntax

Various popular features are available that encode information on syntactic relations of the target word and predicate-argument structures [21]; [22]; [38].

Various studies have been conducted to investigate the effect of individual features, either with the help of automatic feature selection algorithms [30]; [38]; [69]; [9] or manually [43];[45]; [37]; [48].

These features from sense annotated training data are employed by supervised classification algorithms to train a classifier on predicting correct senses to unseen instances. Various supervised machine learning methods are available that can perform the exact way for the features to be used in identifying the correct sense of a target word. Some of the popularly employed ML algorithms in the literature use different underlying classification theories, mostly involving methods based on decision rules, probabilistic approaches, instance- based approaches and support vector machines

Because of their popularity, supervised machine learning methods are typically employed for WSD task. There are various popular supervised ML algorithms available to help in addressing disambiguation of word senses. In contrast to the knowledge-based WSD experiments, supervised machine learning experiments have distinct aspects that need to be related with multiple previous studies. The most similar studies would be the one involving assessment and implementation of a wide range of features including [30], [43]; [45], [37], and [48] or the comparison and evaluation of a wide range of supervised ML algorithms (including [49], [70], [53], [4], [33], and [39]) or those that employ both including [38], [69], [9], [39], and [67]. This section presented the different forms of supervised methods.

2.2 Semi-Supervised

Many word sense disambiguation algorithms use semi-supervised learning which allows both labelled and unlabeled data because of insufficient training data. The Yarowsky (1995) algorithm and (Blum & Mitchell, 1998) were early examples of such algorithms. A semi-supervised model was recommended by Zhao, [72] that used context weighting technique founded on the paradigm of Phrase Structure Tree (PTree) and Dependency Relation Graph (DGraph). With respect to nouns, verbs, and adjectives disambiguation, these techniques result in considerable advances over CW-WSD techniques. For testing the model, all English words of dataset Sensaval-3 were used. Başkaya, [11] offers an all-inclusive study of the structure and valuation of WSD systems. A semi-supervised WSD system was recommended that comprised a small amount of sense-annotated data from Word Sense Induction. The Word Sense Induction is a completely unsupervised methodology that learns the diverse senses of a word based on how it is used without any human intervention. This system is constituted of two components, a Word Sense Induction (WSI) logic and a function that translates the model's sense annotations or explanations into those of a different sense inventory. The suggested WSID system is a heterogeneous ensemble made from the output of all the four WSI models. The outputs of all the WSI systems are combined for every instance and the instance is then identified with the induced senses of all the systems. The mapping function then uses the collective annotations as characteristics to predict the sense. A semi-supervised algorithm was suggested by Jain, [32] for WSD that employed a weighted graph method to determine the intended meaning of a word based on a specific context. The algorithm also used a centrality measure calculation method based on priority that explored the importance of different semantic relations. The algorithm is initiated by accepting a sample text from the user as an input and then identifying the word that requires to be disambiguated. The word identified will be the target word in the algorithm. The sample text also includes the clue word that is required for executing the algorithm. In creating a better WordNet graph, this clue word is used. The SemCor dataset was utilized for implementing the algorithm. Taghipour, [61] studies two different methods of integrating word embeddings in a word sense disambiguation environment, and also assesses these two methods for all-words tasks, some SensEval/ SemEval lexical samples, and also domain-specific lexical sample tasks. A continuous-space demonstration of words or word embeddings was used; as these offer a considerable amount of important information, and thereby improve generalization accuracy. The word embeddings are generally derived from unlabeled data with the help of unsupervised techniques.

semi-supervised techniques Consist of, large amounts of untagged corpora are being used to supply the co-occurrence of information that supplements the tagged corpora. These methods have the potential to assist in the version of supervised models to different domains.

2.3 Unsupervised Approaches

It is known that unsupervised approaches keep away from the knowledge acquisition bottleneck [24], i.e. the extensive resources' poverty that are tagged with word senses manually. Unsupervised approaches to WSD depend on the idea that the same sense of a word has a tendency to have similar neighboring words. Here, input text is used to prompt word senses by clustering word occurrences, after which new occurrences are categorized into prompted clusters. These approaches do not rely on labelled dataset, and they do not take advantage, in their purest version, of any machine-readable resources like dictionaries, thesauri, or ontology. Since these methods do not use any kind of dictionary or other similar resources, they are not able to be dependent on a shared reference inventory of senses. This establishes the primary disadvantage of a fully unsupervised system [50].

While WSD is usually identified as a sense labelling task, i.e. assigning a sense tag to a target word, unsupervised WSD may involve word sense discrimination, i.e. looking to distribute 'word occurrences into many classes by distinguishing any two occurrences regardless of whether they belong to the same sense or not' [60]. Certainly, unsupervised WSD approaches have a different aim when compared to supervised and knowledge-based methods, which detect sense clusters by comparing with the allocation of sense labels. However, both sense labelling and sense discrimination are considered sub-problems in a WSD task [60] and are rather strictly related, where the generated clusters can be used at a later point to sense the occurrences of tag words. Unsupervised approaches for WSD are categorized into three methods, namely word clustering, co-occurrence graphs and methods involving context clustering, and this section covers all the three methods.

Many natural language processing researchers continued experiments with various unsupervised learning algorithms and their applications to word sense disambiguation. [57] follow the footsteps of Schutze with a comprehensive evaluation of the various forms of the context-group discrimination algorithm on the Senseval2 data. [15] describes experiments with clustering of Chinese verbs in a space of rich linguistic features. [5] diverge from the standard vector space model representations in favor of two graph-based algorithms; they experiment with [65] and a form of [14] for unsupervised word sense disambiguation. [42], [65] instead of developing a method for the discrimination of senses, they propose a technique for the automatic detection of the most frequent sense of the word. Because the experiments of McCarthy and colleagues highlight certain points that are important for the motivation of this dissertation proposal, we will look at them more closely.

In automatic word sense disambiguation, the most common-sense heuristic is known to be extremely powerful: because the sense distribution of most words is highly skewed, the most frequent sense baseline beats many supervised systems at Senseval2 [20] even though these systems are trained to take the local context of the target word into account. Even systems that manage to outperform the predominant sense baseline, often back off to the most frequent sense heuristic when they fail to assign a sense with a sufficient degree of confidence. In these systems, the most frequent sense is usually determined from WordNet, which orders senses by frequency of occurrence in the manually tagged corpus SemCor [46].

Much research has been recently devoted to the notion of distributional similarity and its applications. Distributional similarity is a measure of similarity that rates pairs of words based on the similarity of the context they occur in (however context is defined). For instance, two nouns (e.g. "book" and "magazine") that frequently occur as objects of the same verb (e.g. "to read") are considered similar [19]. One application of distributional similarity is in automatic thesaurus generation. A thesaurus generation system outputs an ordered list of synonyms (known as neighbors) ranked by their similarity to the target word. Because the target word conflates different meanings, a list of its automatically generated neighbors will contain words relating to different senses of the target word. The approach to finding the predominant sense for a target word that is taken exploits the fact that the quantity and degree of similarity of neighbors must relate to the predominant sense of the target word in the context from which the neighbors were extracted [42]. In a neighborhood list there will be more words relating to the most frequent sense of the target word and these neighbors will have higher similarity to it in comparison with the less frequent senses. In addition to the automatically generated thesaurus, McCarthy et al. make use of the notion of semantic similarity between senses that can be computed using WordNet similarity package [55]. This latter component is necessary because the words in a neighbor list may themselves be polysemous and a semantic similarity metric is needed to estimate their relatedness to various senses of the target word. To find the predominant sense of a word, each member of its neighbor list is assigned a score that reflects that neighbor's degree of distributional similarity to each of the senses of the target word. These scores are summed up and the sense receiving the maximum score is declared the most frequent.

Zhang in [71], exploit the use of similarity score measurement in an algorithm named genetic word sense disambiguation. This method used [68] similarity measure to calculate the relatedness between each pair of senses, and corpus domain to extract domain terms. This method uses genetic algorithm to explore highest score of relatedness to be disambiguated. [25], make the use of Lesk relatedness measurement to measure the relatedness of an ambiguous word with neighboring words. Because of the high dimensionality of the search space, a genetic algorithm is used to find a near-optimal combination of sense choices. [28] employed a genetic algorithm with semantic relations for WSD. This method exploited semantic relation of WordNet for the sake of finding the most coherent set of senses.

WSD was devised by [52] as a variant to the Travelling Salesman Problem (TSP) to increase the context's general semantic relatedness for disambiguation. Ant colony optimization is a robust nature-inspired algorithm employed in a reinforcement learning manner to address the formulated TSP by integrating similarity measurement based on the Vector Space Model and the Lesk algorithm. In this, the combination of knowledge-based methods was found to be superior when compared with the most frequent sense heuristic and was also observed to considerably minimize the difference between supervised and knowledge-based methods. The resolution of lexical ambiguity is vital to most natural language processing tasks, and as solutions, numerous computational techniques have already been proposed. [16] proposed a method to perform lexical disambiguation of text by utilising the definitions through a machine-readable dictionary combined with the technique of simulated annealing. The method relies on complete sentences for its functioning and simultaneously makes an attempt to select the best combinations of word senses for each word in a sentence. The words in the sentences could be any of the 28,000 headwords in Longman's Dictionary of Contemporary English (LDOCE), which are then disambiguated comparatively based on the senses provided in LDOCE. This fully automatic method does not need hand-tagging of text or hand-coding of lexical entries and based on a sample set of 50 sentences, the results were quite similar to those of other researchers.

For the lexical ambiguity, Rosso et al. (2003) proposed a fully automatic method that employs a wide range of noun taxonomy from the WordNet with the idea of conceptual distance amongst concepts. A formula of conceptual density was developed through this method to aid in lexical disambiguation. This formula can be considered a generalised version of the [6] conceptual density measurement, where numerous refinements were introduced, and an extensive examination was done for all significant combinations. A set of files was selected from SemCor corpus to perform evaluation of the proposed method, which can be considered the sense tagged version of Brown corpus.

2.4 Knowledge-based WSD

As knowledge sources, knowledge-based WSD techniques primarily make use of dictionaries or WordNets to disambiguate amongst word senses. These techniques can also calculate the word overlaps or semantic relatedness amongst different word senses and words to forecast the correct sense for a given context.

Annotated training data is not employed but rather linguistic clues like selection restrictions, overlapping of words with definitions or similarity between two words present in a knowledge base are employed to tackle the WSD task. As these linguistic clues are normally not limited to certain word classes, all words in a running text can be disambiguated through knowledge-based systems. This high coverage is the major advantage of employing knowledge-based systems when compared with the supervised machine learning systems that can be applied only to a restricted set of lemmas that comprise sense-annotated training material. On the other hand, supervised systems generally outpace knowledge-based systems in such cases with a restricted set of lemmas [44]; [41]; [50].

As mentioned earlier, to employ dictionary-based or knowledge-based approach, the similarity between two words in a knowledge base must be measured first for solving the WSD task. Section 3 focuses on the studies and works relevant to the methods involving knowledge-based WSD Semantic similarity measures.

3. Semantic Similarity Measure

The similarity between two terms can be measured through a number of methods, where each method is based on a specific concept. These methods rely on the WordNet's structure to determine a numeric degree that identifies how the two concepts are similar [54]. Physically counting the length between two concepts is the simplest version of these methods. Also, these methods are presented with few limitations provided the path between highly particular concepts shows much smaller distinctions in semantic similarity when compared with the path lengths of very general concepts. The first semantic similarity measure in the biomedical domain was proposed by [44] by employing the path length between biomedical terms in MeSH 4, a medical hierarchy ontology. There are also different variants to the semantic similarity measures proposed by [68].

3.1 Lin Algorithm

A measure was provided by Lin for computing the semantic relatedness of word senses by making use of the concepts' information content (IC) in WordNet (see Eq. 1):

$$sim\ lin(c1,c2) = \frac{2\times IC\left(\left(LCS(c1,c2)\right)\right)}{IC(c1)+IC(c2)} \quad (1)$$

3.2 Wu Palmer Algorithm (WUP)

A measure was developed by Wu & Palmer to estimate the similarity by taking into account the depths of two identified synsets within the WordNet taxonomies, along with LCS' depth (see Eq. 2):

Sim wup (c1, c2) = Log (2×depth (LCS (c1, c2)) / depth(c1) + depth (c 2)) (2)

Equation (2) is used to detect if the WUP's value is greater than zero and less than or equal to one. Since LCS' depth is not detected as zero, the similarity detected value cannot be taken as zero as the taxonomy root's depth is detected as one. Thus, the value was considered to be one in the event where the two similar synset inputs are detected.

3.3 Latent Semantic Analysis (LSA)

Based on [23], LSA starts from the Latent Semantic Indexing (LSI) method to continue with information retrieval. LSI improves the document retrieval process by decreasing the larger-term-by-document matrix to make it fit into a smaller space through singular value decomposition (SVD) [17]. A similar methodology is used by the LSA, even though it consists of a word-by-context presentation.

The LSA employs an M*N co-occurrence matrix to detect a corpus of text, where the M rows are matched based on the word types. The N columns signify a unit of context that could be a sentence, a phrase, or a paragraph. Within the matrix, the individual cells give the count regarding the number of times a word given is identified within a row according to the context within the column.

A few differences exist between the LSA and the LSI, which are mainly based on the context definitions. A document is identified by the LSI, while the LSA, which is regarded to be more flexible, detects a paragraph within the text. If a document is identified by the context unit within the LSA, the LSI as well as the LSA are identified as a single technique. After co-occurrence cell counts are collected and altered, they are transformed in a similar manner, whereby the singular value decomposition (SVD) is employed by the M * N matrix to identify different variations of decomposition.

3.4 Jiang and Conrath Algorithm (JCN)

The semantic relatedness of word senses are calculated through the Jiang and Conrath measure depending on the combination of edge counts in the WordNet "is-a" hierarchy and the WordNet concept's IC values. Values are computed through this measure indicating the semantic distance amongst words in contrast to their semantic relatedness. Thus, to integrate this method, the values have to be inverted for obtaining a measure of semantic relatedness.

3.5 Lesk Algorithm

A word sense disambiguation algorithm was introduced by Lesk [1986], which functions on the assumption that in a text, words that occur together are inclined towards sharing common words in terms of their definitions of a dictionary. The Lesk algorithm functions by assigning sense to an ambiguous target word that has most word overlaps within the definitions of the words in a dictionary based on the context of that target word. This algorithm has two main underlying assumptions (recorded by [8]): (i) in a text, words that occur together were inclined to be used in related senses and (ii) two senses can be termed more related should they have more common words in their definitions.

As a source of sense definitions, although Lesk would use the Oxford Advanced Learner's Dictionary of Current English, however, any semantic resource that could offer sense definitions could be used. For instance, the studies by Kilgarriff [34], [7], [63], [62], [56], [47], as described below, have employed WordNet as a resource of sense definitions.

Two main problems can be related with the original Lesk algorithm: (i) when two or more words are being compared, the number of comparisons also grows exponentially and (ii) most of the dictionary definitions are sparse, which leads to low coverage of the WSD algorithm and insufficient word overlaps. This is the reason for the number of very few studies till date (including [63] and [62], which have applied the Lesk algorithm in its original form.

To address these problems and enhance the performance of algorithm disambiguation, different variants of the Lesk algorithm have been proposed. The simplified Lesk algorithm is the most popular approach to solve the computational complexity problem by comparing more than two words [35], where sense definitions are directly compared with the context (instead of comparing with the sense definitions of each word in the context) to disambiguate each word [44]. This simplified strategy has been applied by almost all studies that employ any variant of the Lesk algorithm. [63] compared various Lesk variants in the all-words dataset of SensEval-2. They found their simplified variant to be much more precise and efficient than the original Lesk algorithm.

The adapted Lesk algorithm is the most common approach to solve the problem of sparse dictionary [7], [8]. The original algorithm is extended through this variant into two main aspects: (i) the overlap calculation involves definitions of related synsets that depend on the underlying idea of two synsets being more related, resulting in more overlaps in their definitions and the corresponding related words' definitions and (ii) higher scores are assigned to overlaps with sequences of words. [7], [8] achieved the second best results with the SensEval-2 lexical sample dataset when compared with the SensEval-2 participating systems. This is almost double the accuracy than the original Lesk algorithm.

Individual disambiguation strategies were used by [10] for each word class and disambiguated adverbs and adjectives through the adapted Lesk algorithm. They achieve higher performance with this word class-specific approach when compared with other applications applying the same WSD algorithm to all word classes.

The original Lesk algorithm has yet another set of proposed variations that compare the way of two sense definitions (or, a sense definition and the context for the adapted Lesk algorithm). Lesk suggested on just counting the number of words in common for the two sense definitions. [35] not only counted the number of common words for the sense definition by considering the context of the target word, but also calculated the sum of each word in common according to the inverse document frequency. The likelihood that a word would occur in an arbitrary sense definition is represented by this inverse document frequency. The cosine similarity between the inverse document frequency vector weighted through the term frequency was computed by [59].

4. Meta-Heuristic for WSD

The WSD task can be solved through the Meta-Heuristic method. The literature contains many published studies related to word sense disambiguation. Also, many approaches are available, including similarity-based methods (depend on thesauri, dictionaries and more generally knowledge sources) and fully supervised methods (employ sense-annotated corpora for training supervised classifiers). Fully supervised methods need large hand-annotated corpora, which is an expensive and rare resource that needs to be custom crafted for a language, sense inventory and even a domain.

However, much research studies have not been carried out for addressing the word sense disambiguation through the use of meta-heuristic approaches. This section presents an assessment of different optimisation algorithms used in word sense disambiguation. [66] conducted an exploratory research on word sense disambiguation. Advanced probabilistic search algorithms, including Bat Algorithm (BA) and a Cuckoo Search Algorithm (CSA), were employed to test experimental data. The two algorithms were then compared with two existing implementations of classical probabilistic optimization algorithms: a Genetic Algorithm and a Simulated Annealing Algorithm. The best configuration score (F1 score) was considered to analyse their efficiency in terms of the function of the number of calls to the scorer (200, 800, 2,000 and 4,000). The algorithm was run 100 times for each algorithm and each scorer call threshold and the average F1 score was plotted across the whole corpus. The 100 runs were then compared to the scoring function's average number of evaluations. [66] use the F1 score to make a comparative assessment with the help of the gold standard of the Semeval 2007 Task 7 WSD task compared with the number of calls to the scoring function. An oracle objective function was employed to analyse the global algorithm's influence on the results, instead of allowing the heuristic scoring function to have an influence. It was concluded that the convergence was too fast as BS stops accepting solutions after some time (inherent to the algorithm, not an explicit convergence criterion). However, the comparison between the two new algorithms, BA and CSA, which were employed first time in WSD, against SA and GA, which were used earlier in WSD, found them to be useable. On the other hand, semantic similarity methods were not observed to help in enhancing WSD's performance in their model. [1], In a lexical substitution setting, proposed extrinsic evaluations of simulated annealing and D-Bees. Each algorithm was employed as the WSD component in the same language-independent, knowledge-based lexical substitution system. German and English datasets were used to test the systems, which exceeded the state-of-the-art performance on the former. Better results were generally associated with the D-Bees system. Then, a few resource specific adaptations were employed depending on the observations of WordNet and GermaNet. These adaptations resulted in significant enhancements of performance for both datasets. The adapted D-Bees system was also examined in a lexical simplification setting, where it was found to surpass simulated annealing performance based on two evaluation metrics. Only when the systems are adapted to the employed linguistic resources or language, the optimal performance could be achieved. This adaptation effort was nonetheless inferior to that needed to source annotated training data required for supervised approaches. [26] stated that the Distributed Arabic Information Retrieval (DAIR) systems' efficiency was enhanced by employing an algorithm called Artificial Bee Colony (ABC) and implementing the query expansion helped in further improving the result quality. While the best relevant document can be explored through the modified ABC (MABC), it can also explore the best synonyms simultaneously for the initially extracted query words from Arabic WordNet (AWN), an external structured resource. Based on the well-known Princeton WordNet (PWN) for English, AWN is considered a free lexical resource for Arabic language. The issues pertaining to the traditional DAIR systems could be solved by using the MABC-SDAIR algorithm. These issues include reduction in the quality of the results because of resource selection, employing ambiguous words in the query and high response time resulting from the use of traditional search with the inverted index.

A model was developed by [27] that employs the fuzzy logic and artificial bee colony (ABC) intelligence to enhance information retrieval systems' performance. A nearest neighbor graph is employed to modify ABC and improve the searching process. However, the model was found to face two problems. The first was the time-consuming issue due to the use of traditional search in the system through the inverted index. The second problem was the reduction in the quality because of the presence of ambiguous words in the query. The proposed system focused on finding the best senses of query words for extracting the best synonyms and used fuzzy logic to tune their weights. Both the content and the lexicon of the document collection were employed to achieve this. The expanded query is generated after the addition of the best synonyms, which helped in improving the quality of the results while the efficiency was increased through the stochastic optimization search of the modified ABC algorithm. Based on the experimental results, the proposed system was found to be superior in terms of recall, precision and latency when compared with the traditional system. [2] proposed a model that employed the D-Bees algorithm along with the knowledge-based unsupervised method to address the problem of WSD task. The bee colony optimization (BCO) is the inspiration behind the design of the D-Bees algorithm, where the problem was solved through the collaboration amongst artificial bee agents. The model was implemented with a dataset called SemEval 2007 coarse-grained English all-words task corpus, which was then compared for genetic algorithms (GA), simulated annealing (SA) and two ant colony optimization techniques (ACO). Better results were achieved through the D-Bees, which outperformed the baseline algorithms, GA and SA.

Several semantic relations were exploited by [71] by employing a genetic algorithm of [68]. However, the window size was used by [71] and [28] to select words for measurement, which would constitute some noisiness in the measured score. For that, dependency relations are employed in this research to solve this problem resulting from parsing operation. This also helped in assigning specific words to be measured with the help of target word, even though the word is far from the target word. Zhang proposed the genetic word sense disambiguation to address the ambiguity issue for nouns only, while Hausman made an attempt to resolve the ambiguity for nouns as well as for other three parts of the speech, namely adjective, verb and adverb.

5. Conclusion

This paper has provided a review for the WSD approaches in which the main types of approaches have been discussed in detail with their corresponding related works. The main types consist of machine learning techniques, semantic similarity measures and meta-heuristic approaches. For the future directions, reviewing the WSD approaches that have been intended to serve specific language would be a great opportunity in the area of Natural Language Processing.

References

[1] Abualhaija, S., Miller, T., Eckle-Kohler, J., Gurevych, I., & Zimmermann, K.-H. (2016). Metaheuristic Approaches to Lexical Substitution and Simplification.

[2] Abualhaija, S., & Zimmermann, K.-H. (2016). D-Bees: A novel method inspired by bee colony optimization for solving word sense disambiguation. Swarm and Evolutionary Computation, 27, 188-195.

[3] Agirre, E., & Edmonds, P. G. (2007). Word sense disambiguation: Algorithms and applications (Vol. 33): Springer Science & Business Media.

[4] Agirre, E., & Martinez, D. (2004). The basque country university system: english and basque tasks. Paper presented at the Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text.

[5] Agirre, E., Martínez, D., de Lacalle, O. L., & Soroa, A. (2006). Two graph-based algorithms for state-of-the-art WSD. Paper presented at the Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing.

[6] Agirre, E., & Rigau, G. (1995). A proposal for word sense disambiguation using conceptual distance. arXiv preprint cmp-lg/9510003.

[7] Banerjee, S., & Pedersen, T. (2002). An adapted Lesk algorithm for word sense disambiguation using WordNet. Paper presented at the International Conference on Intelligent Text Processing and Computational Linguistics.

[8] Banerjee, S., & Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. Paper presented at the IJCAI.

[9] Bas, D., Broda, B., & Piasecki, M. (2008). Towards word sense disambiguation of Polish. Paper presented at the Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on.

[10] Basile, P., De Gemmis, M., Gentile, A. L., Lops, P., & Semeraro, G. (2007). UNIBA: JIGSAW algorithm for word sense disambiguation. Paper presented at the Proceedings of the 4th International Workshop on Semantic Evaluations.

[11] Başkaya, O., & Jurgens, D. (2016). Semi-supervised learning with induced word senses for state of the art word sense disambiguation. Journal of Artificial Intelligence Research, 55, 1025-1058.

[12] Berleant, D. (1995). Engineering "word experts" for word disambiguation. Natural Language Engineering, 1(04), 339-362.

[13] Bhingardive, S., & Bhattacharyya, P. (2017). Word Sense Disambiguation Using IndoWordNet The WordNet in Indian Languages (pp. 243-260): Springer.

[14] Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine.[En línea]. Disponible en Web.

[15] Chen, J., & Palmer, M. (2004). Chinese verb sense discrimination using an EM clustering model with rich linguistic features. Paper presented at the Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.

[16] Cowie, J., Guthrie, J., & Guthrie, L. (1992). Lexical disambiguation using simulated annealing. Paper presented at the Proceedings of the 14th conference on Computational linguistics-Volume 1.

[17] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American society for information science, 41(6), 391.

[18] Dinu, G., & Kübler, S. (2007). Sometimes less is more: Romanian word sense disambiguation revisited. Paper presented at the Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP.

[19] Dligach, D., & Palmer, M. (2008). Novel semantic features for verb sense disambiguation. Paper presented at the Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers.

[20] Edmonds, P., & Cotton, S. (2001). SENSEVAL-2: overview. Paper presented at the The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems.

[21] Fellbaum, C., Palmer, M., Dang, H. T., Delfs, L., & Wolf, S. (2001). Manual and automatic semantic annotation with WordNet. WordNet and Other Lexical Resources, 3-10.

[22] Florian, R., Cucerzan, S., Schafer, C., & Yarowsky, D. (2002). Combining classifiers for word sense disambiguation. Natural Language Engineering, 8(04), 327-341.

[23] Furnas, G. W., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R. A., Streeter, L. A., & Lochbaum, K. E. (1988). Information retrieval using a singular value decomposition model of latent semantic structure. Paper presented at the Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval.

[24] Gale, W. A., Church, K. W., & Yarowsky, D. (1992). A method for disambiguating word senses in a large corpus. Computers and the Humanities, 26(5-6), 415-439.

[25] Gelbukh, A., Sidorov, G., & Han, S.-Y. (2003). Evolutionary approach to natural language word sense disambiguation through global coherence optimization. WSEAS Transactions on Computers, 2(1), 257-265.

[26] Hassan, A. K. A., & Hadi, M. J. (2017a). PROPOSED MABC-SDAIR ALGORITHM FOR SENSE-BASED DISTRIBUTED ARABIC INFORMATION RETRIEVAL. Journal of Theoretical and Applied Information Technology, 95(3), 543.

[27] Hassan, A. K. A., & Hadi, M. J. (2017b). Sense-Based Information Retrieval Using Fuzzy Logic and Swarm Intelligence.

[28] Hausman, M. (2011). A genetic algorithm using semantic relations for word sense disambiguation. University of Colorado at Colorado Springs.

[29] Hoste, V., Daelemans, W., Hendrickx, I., & van den Bosch, A. (2002). Dutch word sense disambiguation: Optimizing the localness of context. Paper presented at the Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions-Volume 8.

[30] Hoste, V., Hendrickx, I., Daelemans, W., & van den Bosch, A. (2002). Parameter optimization for machine-learning of word sense disambiguation. Natural Language Engineering, 8(04), 311-325.

[31] Ide, N., & Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. Computational Linguistics, 24(1), 2-40.

[32] Jain, A., Tayal, D. K., & Vij, S. (2017). A Semi-Supervised Graph-based Algorithm for Word Sense Disambiguation. Global Journal of Enterprise Information System, 8(2), 13-19.

[33] Joshi, M., Pakhomov, S. V., Pedersen, T., & Chute, C. G. (2006). A comparative study of supervised learning as applied to acronym expansion in clinical reports. Paper presented at the AMIA.

[34] Kilgarriff, A., & Palmer, M. (2000). Introduction to the special issue on SENSEVAL. Computers and the Humanities, 34(1), 1-13.

[35] Kilgarriff, A., & Rosenzweig, J. (2000). Framework and results for English SENSEVAL. Computers and the Humanities, 34(1), 15-48.

[36] Kübler, S., & Zhekova, D. (2009). Semi-Supervised Learning for Word Sense Disambiguation: Quality vs. Quantity. Paper presented at the RANLP.

[37] Le, C. A., & Shimazu, A. (2004). High WSD Accuracy Using Naive Bayesian Classifier with Rich Features. Paper presented at the PACLIC.

[38] Lee, Y. K., & Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. Paper presented at the Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10.

[39] Martínez, D. (2007). Supervised corpus-based methods for WSD Word Sense Disambiguation (pp. 167-216): Springer.

[40] Martínez, D., Agirre, E., & Màrquez, L. (2002). Syntactic features for high precision word sense disambiguation. Paper presented at the Proceedings of the 19th international conference on Computational linguistics-Volume 1.

[41] McCarthy, D. (2009). Word sense disambiguation: An overview. Language and Linguistics compass, 3(2), 537-558.

[42] McCarthy, D., Koeling, R., Weeds, J., & Carroll, J. (2004). Finding predominant word senses in untagged text. Paper presented at the Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.

[43] Mihalcea, R. (2002). Instance based learning with automatic feature selection applied to word sense disambiguation. Paper presented at the Proceedings of the 19th international conference on Computational linguistics-Volume 1.

[44] Mihalcea, R. (2006). Knowledge-based methods for WSD. Word Sense Disambiguation: Algorithms and Applications, 107-131.

[45] Mihalcea, R. F. (2002). Word sense disambiguation with pattern learning and automatic feature selection. Natural Language Engineering, 8(04), 343-358.

[46] Miller, G. A., Leacock, C., Tengi, R., & Bunker, R. T. (1993). A semantic concordance. Paper presented at the Proceedings of the workshop on Human Language Technology.

[47] Miller, T., Biemann, C., Zesch, T., & Gurevych, I. (2012). Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation. Paper presented at the COLING.

[48] Młodzki, R., Kopeć, M., & Przepiórkowski, A. (2012). Word Sense Disambiguation in the National Corpus Of Polish. Prace Filologiczne(LXIII), 155-166.

[49] Mooney, R. J. (1996). Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. arXiv preprint cmp-lg/9612001.

[50] Navigli, R. (2009). Word sense disambiguation: A survey. ACM Computing Surveys (CSUR), 41(2), 10.

[51] Ng, H. T., & Lee, H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. Paper presented at the Proceedings of the 34th annual meeting on Association for Computational Linguistics.

[52] Nguyen, K.-H., & Ock, C.-Y. (2013). Word sense disambiguation as a traveling salesman problem. Artificial Intelligence Review, 1-23.

[53] Pedersen, T. (2001). A decision tree of bigrams is an accurate predictor of word sense. Paper presented at the Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies.

[54] Pedersen, T. (2010). Information content measures of semantic similarity perform better without sense-tagged text. Paper presented at the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics.

[55] Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet:: Similarity: measuring the relatedness of concepts. Paper presented at the Demonstration papers at HLT-NAACL 2004.

[56] Ponzetto, S. P., & Navigli, R. (2010). Knowledge-rich word sense disambiguation rivaling supervised systems. Paper presented at the Proceedings of the 48th annual meeting of the association for computational linguistics.

[57] Purandare, A., & Pedersen, T. (2004). Improving word sense discrimination with gloss augmented feature vectors. Paper presented at the Workshop on Lexical Resources for the Web and Word Sense Disambiguation.

[58] Raganato, A., Camacho-Collados, J., & Navigli, R. (2017). Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. Paper presented at the Proc. of EACL.

[59] Ramakrishnan, G., Prithviraj, B., & Bhattacharyya, P. (2004). A gloss-centered algorithm for disambiguation. Paper presented at the Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text.

[60] Schütze, H. (1998). Automatic word sense discrimination. Computational linguistics, 24(1), 97-123.

[61] Taghipour, K., & Ng, H. T. (2015). Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains. Paper presented at the HLT-NAACL.

[62] Torres, S., & Gelbukh, A. (2009). Comparing similarity measures for original WSD lesk algorithm. Research in Computing Science, 43, 155-166.

[63] Vasilescu, F., Langlais, P., & Lapalme, G. (2004). Evaluating Variants of the Lesk Approach for Disambiguating Words. Paper presented at the LREC.

[64] Veenstra, J., Van den Bosch, A., Buchholz, S., & Daelemans, W. (2000). Memory-based word sense disambiguation. Computers and the Humanities, 34(1-2), 171-177.

[65] Véronis, J. (2004). Hyperlex: lexical cartography for information retrieval. Computer Speech & Language, 18(3), 223-252.

[66] Vial, L., Tchechmedjiev, A., & Schwab, D. (2017). Comparison of Global Algorithms in Word Sense Disambiguation. arXiv preprint arXiv:1704.02293.

[67] Wiriyathammabhum, P., Kijsirikul, B., Takamura, H., & Okumura, M. (2012). Applying deep belief networks to word sense disambiguation. arXiv preprint arXiv:1207.0396.

[68] Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. Paper presented at the Proceedings of the 32nd annual meeting on Association for Computational Linguistics.

[69] Yarowsky, D., & Florian, R. (2002). Evaluating sense disambiguation across diverse parameter spaces. Natural Language Engineering, 8(4), 293.

[70] Zavrel, J., Degroeve, S., Kool, A., Daelemans, W., & Jokinen, K. (2000). Diverse classifiers for NLP disambiguation tasks comparisons, optimization, combination, and evolution. Paper presented at the Twente Workshops on Language Technology.

[71] Zhang, C., Zhou, Y., & Martin, T. (2008). Genetic word sense disambiguation algorithm. Paper presented at the Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on.

[72] Zhao, G. Z., & Zuo, W. L. (2014). Semi-Supervised Word Sense Disambiguation via Context Weighting. Paper presented at the Advanced Materials Research.