World Academy of Science, Engineering and Technology
International Journal of Cognitive and Language Sciences
Vol:18, No:1, 2024

Contextual SenSe Model: Word Sense Disambiguation Using Sense and Sense Value of Context Surrounding the Target

Vishal Raj, Noorhan Abbas

***Abstract*—**Ambiguity in NLP (Natural Language Processing) refers to the ability of a word, phrase, sentence, or text to have multiple meanings. This results in various kinds of ambiguities such as lexical, syntactic, semantic, anaphoric and referential. This study is focused mainly on solving the issue of Lexical ambiguity. Word Sense Disambiguation (WSD) is an NLP technique that aims to resolve lexical ambiguity by determining the correct meaning of a word within a given context. Most WSD solutions rely on words for training and testing, but we have used lemma and Part of Speech (POS) tokens of words for training and testing. Lemma adds generality and POS adds properties of word into token. We have designed a method to create an affinity matrix to calculate the affinity between any pair of lemma_POS (a token where lemma and POS of word are joined by underscore) of given training set. Additionally, we have devised an algorithm to create the sense clusters of tokens using affinity matrix under hierarchy of POS of lemma. Furthermore, three different mechanisms to predict the sense of target word using the affinity/similarity value are devised. Each contextual token contributes to the sense of target word with some value and whichever sense gets higher value becomes the sense of target word. So, contextual tokens play a key role in creating sense clusters and predicting the sense of target word, hence, the model is named Contextual SenSe Model (CSM). CSM exhibits a noteworthy simplicity and explication lucidity in contrast to contemporary deep learning models characterized by intricacy, time-intensive processes, and challenging explication. CSM is trained on SemCor training data and evaluated on SemEval test dataset. The results indicate that despite the naivety of the method, it achieves promising results when compared to the Most Frequent Sense (MFS) model.

***Keywords*—**Word Sense Disambiguation, WSD, Contextual SenSe Model, Most Frequent Sense, part of speech, POS, Natural Language Processing, NLP, OOV, out of vocabulary, ELMo, Embeddings from Language Model, BERT, Bidirectional Encoder Representations from Transformers, Word2Vec, lemma_POS, Algorithm.

I. Introduction

WORD form is defined in terms of orthography and phonological form. Words having the same orthography with different meanings are called homograph such as “bank” in “bank of river” or “bank of America”. Homographs can be pronounced differently (e.g., “bass the fish” and “bass the instrument”, here “bass” is pronounced differently) or the same (e.g., fair meaning “equitable” or “a carnival”, here “fair” is pronounced same). Words having different orthography but same phonological form are called homophones like too/two or write/right. Some words, like bark, fall into more than one category — bark on a tree and bark of a dog are both homophones (sounding the same) and homographs (being spelled the same). The word homonyms are often used to refer to all such words (homographs and homophones). Homonyms have completely different meanings and different origins. Similarly, there is Polysemy, in which words have the same form, but slightly different meaning (e.g., He drank a glass of milk/He forgot to milk the cow). Polysemy has a different yet related meaning. Polysemy has related word origins. Both Homonyms and Polysemous led to the problem of WSD in NLP.

Current WSD Models which are trained on words have the problem of out of vocabulary (OOV) while determining the sense of word on test set. We have addressed this matter by employing lemmatization; however, this approach introduces an excessive degree of generality into the model. So, we have added properties of lemma using POS tag and tokenized both train and test set into lemma_POS tokens. The approach of lemma_POS tokens has significantly improved the performance of the model. After tokenization, we have designed formulae suitable for lemma_POS tokens to find affinity value between two tokens and created an affinity matrix with newly designed formula. Sense clusters of all contextual tokens of a target word are created based on their mutual affinity with each other. We have devised three distinct mechanisms for predicting the sense of the target word through the utilization of sense clusters.

The first mechanism compares the affinity of a context token with all sense clusters of the target token to judge the sense and affinity value of the contextual token. The second mechanism uses a cloud of context to judge the sense and affinity value of a contextual token. The third mechanism uses the word vectors of lemma_POS tokens to judge the sense and similarity value of a contextual token. Finally, summing up the sense contribution of all contextual tokens of target word is used to determine the final sense of the target word. The whole process of creation of sense clusters and prediction of sense of target word is called Contextual SenSe Model (CSM). Our CSM has shown 30-35% higher accuracy on train set (SemCor) and 20% higher accuracy on test set (SemEval) in comparison to MFS to solve WSD.

By disambiguating words, WSD helps improve the accuracy

Vishal Raj is doing Masters in AI from University of Leeds, School of Computing, Leeds, UK (e-mail: Od20vr@leeds.ac.uk).

Noorhan Abbas is a Teaching Fellow at University of Leeds, School of Computing, Leeds, UK (e-mail: N.H.Abbas@leeds.ac.uk).

of downstream NLP tasks such as information retrieval [1], machine translation [2], sentiment analysis [3], question answering [4] and text summarization [5]. So, WSD contributes to the broader goal of enhancing natural language understanding by addressing lexical ambiguity.

A. Background and Related Work

Lesk Algorithm [6] is a dictionary-based method that uses word definitions to disambiguate word senses. It selects the sense of a target word based on the overlap of words in the definitions of the target word and its surrounding context.

Personalized PageRank [7] uses a graph-based algorithm that leverages semantic resources like WordNet [8] to measure the relatedness between word senses and their context. Extended Lesk [9] modifies the original Lesk algorithm by incorporating semantic relations from WordNet and other lexical resources to improve disambiguation performance.

Hyperlex performs graph based WSD using co-occurrence graphs. A graph is built for each target word in corpus where nodes represent content words co-occurring with the target word in context, and edges connect the words which co-occur in these contexts. The second step iteratively selects the node with highest degree in the graph (root hub) and removes it along with its adjacent nodes. Each such selection corresponds to isolating a high-density component of the graph, in order to select a sense of the target word. In the last step, the root hubs are linked to the target word and the Minimum Spanning Tree (MST) of the graph is computed to disambiguate the target word in context [10]. We have created an affinity graph using lemma_POS tokens instead of words for WSD using different formulae and mechanism.

Early supervised learning methods for WSD involved training classifiers using hand-labeled datasets. These classifiers often used features such as POS tags, local context words, and syntactic relations [11]. Latent Semantic Analysis [12] is an unsupervised method that uses dimensionality reduction to map words and their contexts into a low-dimensional semantic space, facilitating sense distinction. Bootstrapping [13] is a semi-supervised approach that iteratively refines a small set of labeled data using a larger, unlabeled dataset to improve WSD performance. Gaustad [14] has used the lemma-based approach for WSD for Dutch language by creating classifiers based on lemma. Further features used in classifier for disambiguation are POS of target word, its left context and right context. Our supervised model has also used lemma, POS and its context for clustering and disambiguation of target word.

Niu et al. [15] used Context Clustering Based on Pairwise Context Similarities for WSD. Our approach uses clusters of lemma_POS (e.g. long_ADJ) tokens and tokens of clusters are decided based on their affinity value to each cluster under hierarchy of POS of lemma. Nameh et al. [16] used Context Similarity to predict sense for WSD. After clustering, we have used affinity value or cosine similarity to predict the sense of target word under disambiguation.

Word Embeddings [17], [18] such as Word2Vec and Global Vectors for Word Representation (GloVe), represent words as dense vectors that capture semantic and syntactic information. We have also used embeddings in one of our prediction algorithms; these embeddings are not of words, but are created using lemma_POS tokens (e.g., long_ADJ).

Contextualized Embeddings [19], [20] include models like ELMo and BERT which generate contextualized embeddings that consider the context in which a word appears, offering improved WSD performance over non-contextualized word embeddings. Contextual Word Disambiguation Using Word Embeddings and WordNet [21] presents a context-based word disambiguation approach that leverages word embeddings and WordNet. These approaches also incorporate word context; however, they are characterized by complexity, time-intensive procedures, and challenging explication.

We have used affinity matrix/word vectors of lemma_POS tokens to capture the contextual relationship between tokens and sense inventory in a training set in a simple and innovative manner.

II. Design and Methodology

A. Core Concepts

Affinity Graph: An affinity graph is created with vertex V and edge E, where V represents the words in text and E are added if the words co-occur in the relation according to syntax in the same paragraph or text. For a given target word, first, the graph is created and then an affinity matrix for the graph is created.

Each edge of the graph is assigned a weight which is the co-occurring frequency of those two words. Weight for edge {m, n} is given by the formula:

$$w_{mn} = \max(1, \text{count}(w_{mn})/\min(\text{count}(w_m), \text{count}(w_n)) \text{ if } m! = n \quad (1)$$

$$w_{mn} = 1 \text{ if } m=n \quad (2)$$

Affinity is defined by the size of the context window around target word. W_{mn} has a value near to one if two words are co-occurring frequently in a given context window. Whereas, the W_{mn} value reaches to zero if two words are co-occurring rarely in a given context window.

Our approach relies on the fact that words co-occurring within a context window and having an affinity value above the threshold belong to one sense and are part of one cluster. Similarly, we can create clusters of words of different senses.

Cluster Creation Based on Lemma and POS Tagging: The challenge in NLP lies in the insufficiency of available data to enable learning from scratch the equivalency of inputs that may appear superficially dissimilar or the critical distinctions within inputs that may appear superficially alike. Hypothetically, the performance of algorithms would significantly improve with access to larger datasets and increased computing power, provided the task must be completed within a reasonable timeframe. However, the challenge remains that we currently lack access to such extensive datasets.

Hence, our approach is based on the premise that executing

two preprocessing tasks, namely lemmatization and POS tagging prior to clustering, can yield highly favorable results with reduced data requirements and time expenditure.

In natural languages, the same word can appear in texts with different forms. A single word "love" can appear in three different forms: "love", "loves" and "loved". Creating different clusters for different word forms will only add complexity and requires training data having all word forms. Lemmatization can bring different forms into one lemma. So, a word form which is not present in the training data can also be disambiguated using its lemma.

The second technique we propose is POS tagging, which takes all the words in a text and tags them with its corresponding POS (its grammatical category, like "noun", "verb", "adjective", "preposition" and so on). By applying this process, the verb "like" has become distinguishable from the preposition "like".

Using existing services of lemmatization and POS tagging in NLP crucially reduces the amount of data and computing power needed to reach the desired results in WSD. Lemmatization streamlines the process of comprehending the collective significance of all instances of a given word, while POS tagging serves to disambiguate words that share a similar form but possess distinct roles in influencing the resolution of the task.

Clusters are created from the context surrounding a word. In this study, we convert words of the training set in the form of lemma_POS tokens. Different clusters are created under POS of lemma of disambiguated word from training set using affinity graph. Tokens falling within instances of a particular sense of a given POS of a lemma are grouped together to constitute a single cluster corresponding to the lemma's POS. So, we create different clusters of lemma_POS tokens under POS of lemma for our complete training set depending on number of senses present under POS of lemma in training set. See Fig. 1 for more clarity on this subject.

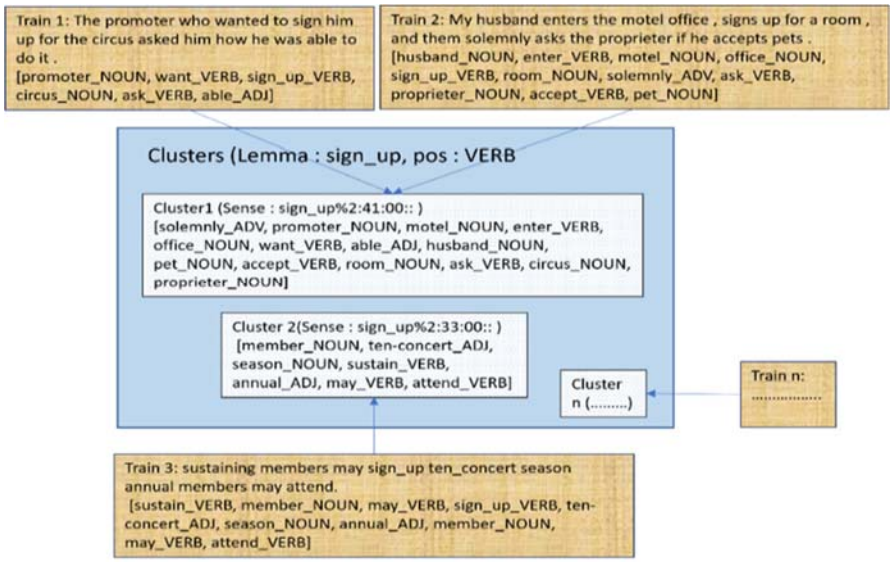

Fig. 1 SenSe Clusters under hierarchy of lemma and POS of target word

Word2vec (lemma_POS token embeddings): Mikolov et al. [17] designed two simple methods for learning continuous word embeddings using neural networks based on Skip-gram or Continuous-Bag-of-Word (CBOW) models and labelled the approach Word2Vec. Word vectors created using these methods map words to points in space that effectively encode semantic and syntactic meaning despite ignoring word order information. Subsequent work leveraging such neural word embeddings has been proven effective on a variety of natural language modeling tasks [22], [23]. These word embeddings do not incorporate the generality of lemma and properties of lemma such as noun, adjective, verb, and adverb, etc. In the proposed algorithm, word embeddings of lemma_POS tokens are created from the training dataset. So, our word embeddings incorporate the generality and properties of lemma.

B. Dataset

Assessing the efficacy of word disambiguation systems presents an enduring challenge, marked by the difficulty of substantiating their enhancements. To address this issue, we have chosen datasets from SemCor/SemEval for both training and evaluation of our models. Our aim is to demonstrate the advancement of the proposed model in contrast to the most frequent sense model. Employing the training set, we have constructed sense clusters for disambiguating words and subjected them to testing using dedicated test sets. Our current focus is primarily on clusters associated with ambiguous words falling under the categories of nouns, verbs, adverbs, and adjectives.

Train Dataset: SemCor [24] is a manual sense-annotated corpus divided into 352 documents for a total of 226,040 sense annotations. SemCor is the largest corpus manually annotated with WordNet senses, and is the main corpus used in the literature to train supervised WSD systems [25], [26].

Test Dataset: We will use five WordNet sense annotated corpora for evaluation of the proposed model, as each one is

having different domain of documents, POS tags and sense inventory. Moreover, these datasets are also used by other researchers for evaluation and will enable them to make a fair comparison with our system. Senseval-2 [27] dataset consists of 2283 sense annotations, including nouns, verbs, adverbs, and adjectives. Senseval-3 [28] dataset is divided into three documents from three different domains (editorial, news story and fiction), totaling 1850 sense annotations. SemEval-07 [29] is the smallest among the five datasets, containing 455 sense annotations for nouns and verbs only. SemEval-13 [30] dataset includes 13 documents from various domains. In this case the original sense inventory was WordNet 3.0, which is the same that we use for all datasets. The number of sense annotations is 1644, although only nouns are considered. SemEval-15 [31] is the most recent WSD dataset available to date and it consists of 1022 sense annotations in four documents coming from three heterogeneous domains biomedical, mathematics/computing, and social issues.

C. Design

Data Preparation Engine: Punctuation and stop words are removed from the training and test datasets. Words are converted to lower case. The training data and test data are both lemmatized and POS tagged using genism library. The remaining words of sentences are converted into lemma_POS tokens. Sentences having length fewer than thresholds are removed from the train and test datasets, as they have very small context for creating clusters and prediction of sense of the target word under disambiguation.

Vocabulary and Affinity Matrix: The unique list of lemma_POS tokens in the dataset become part of the vocabulary. Each token in the vocabulary is assigned an index. The affinity matrix is created with DxD dimensions, where D is the size of vocabulary. For any pair of tokens, the context window defines the affinity relation between them. Each entry in the matrix is a count of affinity of two tokens in the training dataset, and the affinity count is divided by the minimum count of both the tokens in the dataset. Further, each entry of the affinity matrix is capped at one. This is done to normalize the matrix. This formula has shown very good accuracy results on the train/test sets. If m and n represent indexes in vocabulary and w_{mn} represents the entry in the m^{th} row and n^{th} column of affinity matrix, then (3) defines the affinity value between two tokens of vocabulary.

$$\text{Affinity value between two tokens} = w_{mn}\,(\text{entry in affinity matrix}) \quad (3)$$

Training Engine (CSM): Contextual words are up to the length of context windows around the target word under disambiguation. Each target word under disambiguation converted into a lemma_POS token and its context words are also converted into lemma_POS tokens. Hierarchy of lemma -> pos -> sense is created for all sense tagged words. Now, for each sense under POS of lemma present in the training data, all contextual tokens of that sense (based on the minimum threshold value of the affinity matrix) are put under each sense as a sense cluster. Remaining contextual tokens of sense are put into the respective sense cluster depending on their maximum affinity toward a cluster. Affinity of contextual token toward any given sense cluster is equal to the sum of affinity of the context token with tokens of a given sense cluster. The number of tokens of a sense cluster to be considered for calculating affinity with a context token is a hyper parameter.

Prediction Algorithm: In this research study, we have designed three different prediction algorithms with three different mechanisms. All three prediction algorithms rely on one common fact, which is that, each contextual token of a target word under disambiguation contributes to the sense with some value. If we sum up the values of all contextual tokens toward all senses of the target token, then, whichever sense gets the highest value, becomes the sense of the target word under disambiguation. This approach evidently shows that all contextual tokens contribute toward the real sense of the target word. That is the reason for labelling the model CSM (Contextual SenSe Model).

As shown in Fig. 2, Algorithm 1 (mechanism 1 - co-occur) uses the affinity between the context token and sense clusters of target tokens to determine the real sense and sense value of that contextual token. Algorithm 2 (mechanism 2 - cloud context) uses the affinity between the target token and cloud of contextual token to determine the real sense and sense value of that contextual token. Algorithm 3 (mechanism 3 - Word2Vec) uses the similarity between the context token and sense clusters of target tokens to determine the real sense and sense value of that contextual token.

III. Experimental Results and Discussions

A. Rationale/Results of Affinity Formulae

Division of Affinity by Minima: Division of affinity of two words in a given context window is done by the minima of individual count of two words instead of using the maxima or multiplication of their counts. For instance, in the sentence: “I saw Pemberton and he is the most insignificant puke I ever saw.” The word “puke” occurred once in the training set, whereas “see” occurred many times. Division by the max/multiplication of their counts would result in a notably diminished value. Accuracy over the training/testing set is also not satisfactory using max/multiplication. Accuracy results achieved on the train/test set using minima are significantly superior to those obtained using maxima or multiplication.

Maxima of Affinity: This can be explained with the following sentence in the training set: “I saw Pemberton and he is the most insignificant puke I ever saw.” This sentence will be tokenized into “['see_VERB', 'person_NOUN', 'insignificant_ADJ', 'puke_NOUN', 'ever_ADV', 'see_VERB'] using the “data preparation engine” of Section II *C.* In this tokenized list, the co-occurrence of 'puke_NOUN' and 'see_VERB' is two, whereas the count of 'puke_NOUN' is one. This causes the affinity to have a value more than one. Capping the maxima of the affinity matrix to one has shown better results.

Affinity of Word with Itself: Affinity is calculated by checking the affinity of a context token with the tokens in

clusters of the target token. If the same context token appears in the cluster of the target token, then its value is set to 1. This has also improved the accuracy of our CSM algorithms.

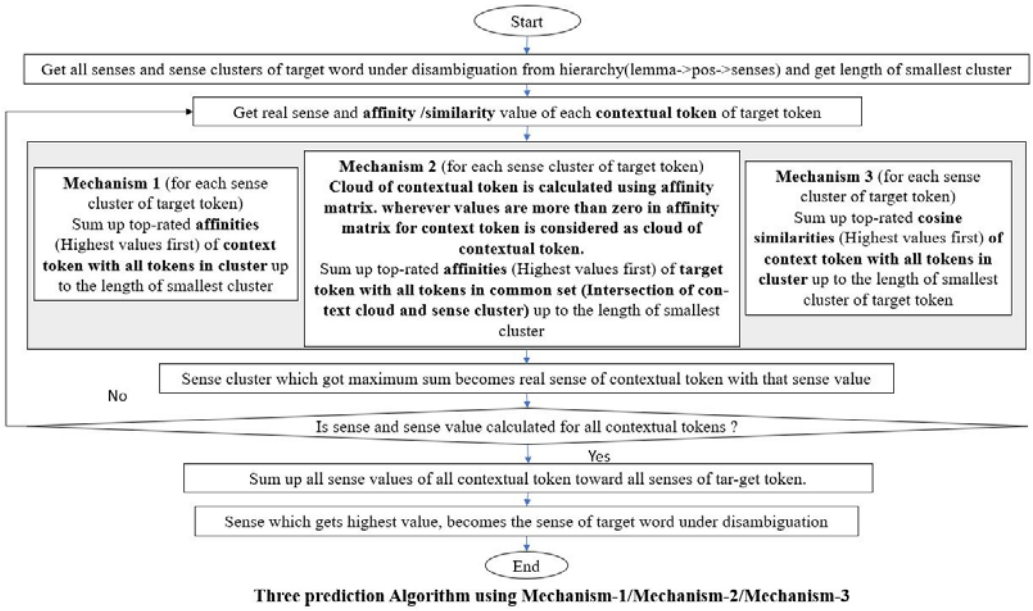

Fig. 2 Three prediction algorithms to judge the sense of a target token (using the sense and sense value of contextual tokens around target token)

B. Accuracy Metric

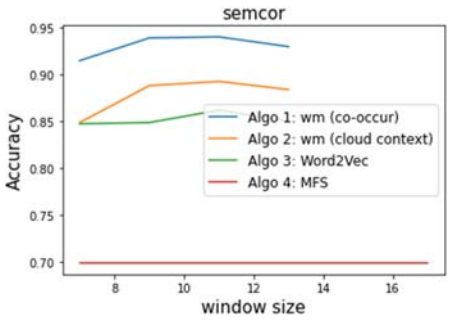

Fig. 3 Comparison of the accuracy of the three prediction algorithms and MFS vs. window size over the training set (SemCor)

Results on the Training Set: As shown in Fig. 3, the accuracy of our three prediction algorithms and MFS are compared with each other over the size of the window on the training dataset. The window size is the context of window around the target token. The horizontal line represents the accuracy of MFS (Most Preferred Sense based on Lemma) algorithm. All our prediction algorithms have shown better performance than MFS on all window sizes. The improvement of our algorithms is approximately 30-35% more compared to MFS. This clearly indicates that the context in which a target token appears has an influence on the sense of the target token. Moreover, it also emphasizes our core concept that, the affinity value of all senses of the target token is calculated by the sum of the contribution of the context tokens towards all senses of target tokens. Whichever sense got the maximum value becomes the sense of the target token.

Results on the Test Set: As shown in Fig. 4, the accuracy of our three prediction algorithms and MFS are compared with each other over the size of window on five test datasets. Here also, all our designed algorithms have shown better performance than MFS on all window sizes. The improvement of our algorithms is approximately 20% compared to MFS.

C. Hyperparameters

Context Window (Affinity Matrix): The context window is defined as the size around a target word including the target word. So, a context window of 11 means five words each to the left and to the right of the target word. If the sentence length is smaller than the context window, then all sentence tokens are considered. If the left context of the target word is less than half of the context window, then the remaining tokens are taken from the right side of the target word and vice-versa.

In Figs. 1 and 2, the accuracy improves when the context window size increases from five, reaching its maxima at 11, before it starts decreasing. This behavior has been overserved in both the training and test sets.

Word vectors are created using either Skip-gram or Continuous Bag-of-Words models. Input parameters and their values are vector_size = 300, min_count = 1, window = 5, negative = 5. Here, a window of five means the total window size around the target token is 11. These parameters achieved the maximum accuracy for WSD.

Length of Cluster (Calculation of Affinity): In WSD, sense clusters can have different lengths. The number of tokens in a sense cluster, used to calculate the affinity of a context token toward each cluster, is a hyperparameter. It has been found that the length of the smallest cluster is the best number to calculate the affinity of each cluster with the contextual word. Whichever cluster has the maximum affinity becomes the sense, and the

sense value of the contextual token is used for disambiguating the sense of the target word.

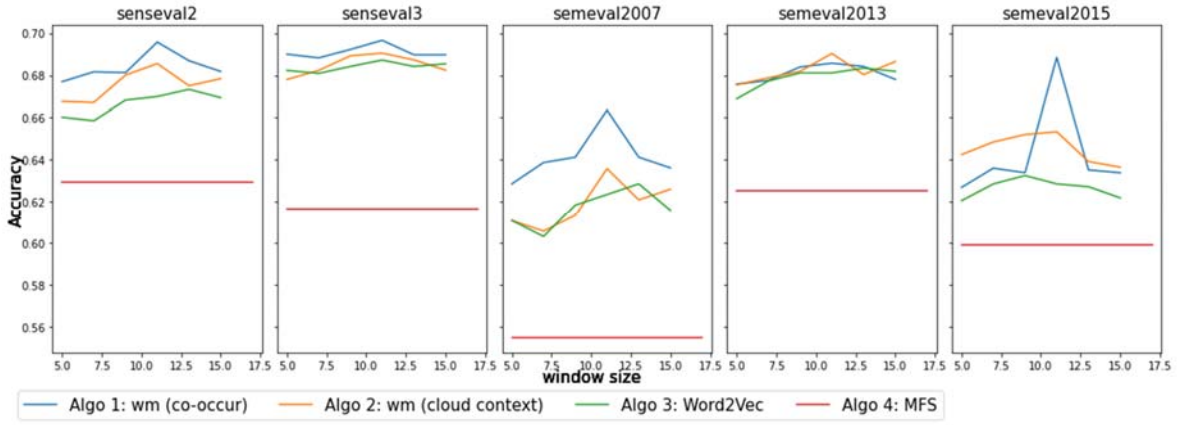

Fig. 4 Accuracy (Three CSM algorithms and MFS) vs. window size over the five test sets

D. Performance Comparison

Lemma and POS Feature: Using context words directly to create sense clusters is too strict criterion, as words in the test set might not appear in training set due to being OOV. Using the lemma of a word improves OOV words, and POS features restrict the generality of a word too much for WSD. Using both features has improved the performance of all our models.

Comparison of Results of the Training/Test Set: Contextual words of target word under disambiguation are used to create sense clusters of target word, and the training data are used for that purpose. Hence, tokens of our sense clusters come from the training set, meaning we have an overreliance on tokens of the training set. We have tried to minimize this overreliance on the training set using lemma and POS of the word, as lemma adds generality and POS add characteristics, which helps in handling of OOV words also. However, this explains the fact that training set has an accuracy of 20-25% more than the test set. This also brings up one more point that we can increase the accuracy of the test set by increasing the size of the training set.

Comparison Based on Prediction Algorithms: The accuracy of Algorithm 1, which calculates the affinity between the context token and sense clusters of the target token works slightly better than Algorithm 2, which uses the context cloud, as the context cloud brings more generality. The accuracy of Algorithm 3, which uses word vectors, is slightly below the other two models. This may be due to the fact that word vectors are not trained well, and the performance of this algorithm can be achieved at par/better in comparison to the other two algorithms by fine-tuning of word vectors.

E. Complexity

Affinity is calculated by running a loop over all contexts around each token. This causes time complexity of O (size of context * total tokens). Normalizing of the affinity matrix has a time complexity of O(n^2), as each entry is accessed to normalize it within a range of 0 to 1. So, the time complexity of the affinity matrix is O(n^2).

Creation of sense clusters requires looping over all tokens of senses of POS of lemma. So, the time complexity of this algorithm is O (number of lemmas * number of POS* number of senses * number of tokens in each sense).

Time complexity of prediction is O (number of contextual tokens * number of sense clusters * number of tokens in each sense).

All the aforementioned results demonstrate that the time complexity of our training and testing algorithm is notably lower when contrasted with deep learning algorithms. In conclusion, these algorithms prove to be highly efficient in achieving commendable accuracy by adeptly comprehending the contextual cues surrounding all target tokens.

IV. Conclusions, Future Work and Ethical Issues

A. Conclusions

In this research study, we have used lemma_POS tokens instead of words for creation of clusters that are created under the hierarchy of lemma and POS. We possess well established libraries such as Genism for doing the lemmatization and POS tagging in the training/test sets. Further word vectors are also created using lemma_POS tokens. This simple approach has captured the generality of lemma so nicely that we reduced the OOV words in the test set without losing the properties of the lemma, where the vocabulary is made using words in the training set. This has yielded a high level of accuracy on the training/test sets in comparison to the MFS model.

Furthermore, we have derived the most optimal affinity formulas for constructing the affinity matrix, and this approach adeptly captures the syntactic and semantic relationships among words within the training corpus. As a result, we have been able to create sense clusters in a straightforward and highly efficient manner.

Two different ways of using the syntactic and semantic relationship of words present in training corpus through affinity matrix and word vectors have been devised. We have designed two prediction algorithms using context and cloud of context for prediction through affinity matrix and one algorithm using

cosine similarity of word vectors. All three of them have a novel, simple and efficient approach in prediction of sense compared to complex and time-consuming deep learning models.

B. Future Work

Ensemble: Our first algorithm compares the affinity of the context token with sense clusters of the target token, the second one uses the cloud of the context token, and the third algorithm uses embedding to check the context similarity with sense clusters of the target token. All our prediction methods are very different from each other and each one has accuracy of more than 50%. So, the combined accuracy of an ensemble of such classifiers is often significantly greater than that of an individual classifier. Our accuracy on the training set is 15-20% higher than the test set, which means slight overfitting of the model. An ensemble of these methods will help to overcome over fitting and underfitting [32].

Recursion and Order of Context Around Token: The accuracy of our model depends on the basis that words present in the test set must be present in the training set and we have tried to achieve it using lemmas. However, there are still cases where words in the test set do not correspond to a token in the training set; in that case, cloud of context (using affinity matrix) can be used to find its correlation with sense clusters of the target token. This approach can be recursively applied to reach the sense contribution of the context token toward the sense clusters of the target word. Our approach uses the context around the target word without considering their order. There are cases where the order can change the sense of the target word. We would like to capture this relationship in a distance matrix and want to use the affinity and distance matrix together for creation of the cluster and in prediction of the sense of the target word.

C. Ethical/Legal Issues

The WSD system can advertently perpetuate biases if the training data contain biased language or societal biases. These issues can be solved by implementing safeguards such as data curation, bias detection and mitigation techniques. The WSD system may struggle in disambiguating words from different cultures, dialects or languages, particularly in international or multicultural contexts. As the WSD system relies on large and diverse training sets to generalize well, we can add a wide range of context or language variation etc., to handle these situations. Many WSD models, particularly deep learning models, can be challenging to interpret and explain. But our model CSM is simple, easy to interpret and explain. It promotes transparency and accountability in design and deployment of WSD systems.

WSD relies on training datasets, so issues related to trademark, copyright and intellectual property rights can be avoided by taking proper authorization for training datasets.

REFERENCES

[1] Zhi Zhong and Hwee Tou Ng. 2012. Word Sense Disambiguation Improves Information Retrieval. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Vol. 1, pages 273–282.

[2] Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word Sense Disambiguation Improves Statistical Machine Translation. Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 33–40.

[3] Hung, C., & Chen, S.-J. 2016. Word sense disambiguation-based sentiment lexicons for sentiment classification. Knowledge-Based Systems, Vol. 110, pages 224-232.

[4] Hung, Jason & Wang, Ching-Sheng & Yang, Che-Yu & Chiu, Mao-Shuen & Yee, George. 2005. Applying Word Sense Disambiguation to Question Answering System for e-Learning, 19th International Conference on Advanced Information Networking and Applications, Vol. 1, pages 157 – 162.

[5] Rahman, N., Borah, B. 2020. Improvement of query-based text summarization using word sense disambiguation. Complex & Intelligent System, Vol. 6, pages 75–85.

[6] Lesk, M. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation, pages 24-26.

[7] Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for Word Sense Disambiguation. Proceedings of the 12th Conference of the European Chapter of the ACL, pages 33–41.

[8] George A. Miller. 1995. WordNet: a lexical database for English. Communication of the ACM, Vol. 38, pages 39–41.

[9] Satanjeev Banerjee and Ted Pedersen. 2002. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet, Lecture Notes in Computer Science, Vol. 2276, Pages 136–145.

[10] Silberer, C. and Ponzetto, S. P. 2010. UHD: Cross-Lingual Word Sense Disambiguation Using Multilingual Co-occurrence Graphs. Proceedings of the 5th International Workshop on Semantic Evaluation, ACL, pages 134–137.

[11] Hwee Tou Ng and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, pages 40- 47.

[12] Hinrich Schütze. 1998. Automatic Word Sense Discrimination. Computational Linguistics, Vol. 24, No. 1, pages 97–123.

[13] David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, pages 189–196.

[14] Tanja Gaustad. 2004. A Lemma-Based Approach to a Maximum Entropy Word Sense Disambiguation System for Dutch. Proceedings of the 20th International Conference on Computational Linguistics, pages 778–784.

[15] Cheng Niu, Wei Li, Rohini K. Srihari, Huifeng Li, and Laurie Crist. 2004. Context clustering for Word Sense Disambiguation based on modeling pairwise context similarities. Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (ACL), pages 187–190.

[16] Nameh, M. S., Fakhrahmad, M., Jahromi, M.Z. 2011. A New Approach to Word Sense Disambiguation Based on Context Similarity. Proceedings of the World Congress on Engineering, Vol. 1, pages 456-459.

[17] Mikolov, Tomas & Chen, Kai & Corrado, G.s & Dean, Jeffrey. 2013. Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. 2013, arXiv preprint arXiv:1301.3781.

[18] Pennington, J., Socher, R., Manning, C.D., 2014. ‘‘Glove: Global vectors for word representation,” in Empirical Methods in Natural Language Processing, pages 1532–1543.

[19] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pages 2227–2237.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pages 4171–4186.

[21] Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Vol. 1, pages 99–110.

[22] Rami Al-Rfou’, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed Word Representations for Multilingual NLP. Proceedings of

the Seventeenth Conference on Computational Natural Language Learning, pages 183–192.

[23] Danqi Chen and Christopher Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 740–750.

[24] George A Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G Thomas. 1994. Using a semantic concordance for sense identification. Proceedings of the workshop on Human Language Technology, pages 240–243.

[25] Eneko Agirre, Oier Lopez de Lacalle, Christiane Fellbaum, Shu-Kai Hsieh, Maurizio Tesconi, Monica Monachini, Piek Vossen, and Roxanne Segers. 2010. SemEval-2010 Task 17: All-Words Word Sense Disambiguation on a Specific Domain. Proceedings of the 5th International Workshop on Semantic Evaluation, pages 75–80

[26] Zhi Zhong and Hwee Tou Ng. 2010. It Makes Sense: A Wide-Coverage Word Sense Disambiguation System for Free Text. Proceedings of the ACL 2010 System Demonstrations, pages 78–83.

[27] Philip Edmonds and Scott Cotton. 2001. Senseval-2: Overview. Proceedings of The Second International Workshop on Evaluating Word Sense Disambiguation Systems, pages 1–6.

[28] Benjamin Snyder and Martha Palmer. 2004. The English all-words task. Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, pages 41–43.

[29] Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pages 87–92.

[30] Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. Proceedings of SemEval 2013, pages 222–231.

[31] Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 288-297.

[32] Cucerzan, R.S., C. Schafer, and D. Yarowsky. 2002. Combining classifiers for word sense disambiguation. Natural Language Engineering, Vol. 8, No. 4, pages 327- 341.