[1]Dr. Rupesh Mahajan

[2]Dr. Chandrakant Kokane

[3]Kishor Pathak

[4]Dr. Manohar Kodmelwar

[5]Kapil Wagh

[6]Mahesh Bhandari

Effect of Supervised Sense Disambiguation Model Using Machine Learning Technique and Word Embedding in Word Sense Disambiguation

Abstract: - Natural language processing includes a subfield called word sense disambiguation, which focuses mostly on words that might have several meanings. Polysemous terms are also referred to as confusing phrases in some circles. The performance of word sense disambiguation depends on how effectively the ambiguous word is recognized by the machine. The discussed word embedding model for the ambiguous words represents the words from the document space to vector space with no data loss. The most identified challenge of ambiguous word representation is the features. The selection and representation of ambiguous words with respect to the features is the tedious task of word embedding. The discussed word embedding model uses countable features of available context for disambiguation. The proposed model is implemented for ambiguous words with context information. The available context of ambiguous/polysemous words is used for disambiguation. The unavailability of the context is the challenge in this model. The Recurrent Neural Network with Large Small Term Memory is used for the classification. The output of the RNN-LSTM is the sense values which are further mapped with the freely available lexical resource WordNet for retrieving the correct sense(meaning).

General Terms: Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

Keywords: Lexical Ambiguity; Word Vector; Word Embedding; Word Sense Disambiguation

I. INTRODUCTION

Word Sense Disambiguation (WSD) is the hot research area of Natural Language Processing (NLP), most of the Artificial Intelligence(AI) based applications are now comfortable with the NLP. Nowadays people are communicating with smart devices and with respect to the commands the devices are functioning. Today machines are accepting the query from the user and the results are generated the results are in the form of information or data but in the future machines will accept inputs from humans and will generate the results here the results will in either in the form of data or actions. This point states the importance of machine translation and the future of machine translation. The performance of any machine transition depends on the WSD: if the machine is not able to recognize the user query it will generate ambiguous results. Word embedding is by far the most important factor in word sense disambiguation (WSD). In the field of natural language processing (NLP), word embedding is an essential technique that attempts to represent words in a vector space that is continuous. It is able to improve a range of tasks associated with natural language processing, such as sentiment analysis, machine translation, and the identification of named entities, which has been contributing to its stratospheric increase in popularity over the past few years. Other

[1] Dr. D.Y. Patil Institute of Technology, Pimpri, Pune, Maharashtra, India, mhjn.rpsh@gmail.com 0009-0004-5371-8080

2Nutan Maharashtra Institute of Engineering and Technology, Talegaon, Pune, Maharashtra, India, cdkokane1992@gmail.com 0000-0001-7957-3933

3Vishwakarma Institute of Information Technology, Pune, Maharashtra, India, kishor.pathak@viit.ac.in 0000-0001-8409-7433

4Vishwakarma Institute of Information Technology, Pune, Maharashtra, India, manohar.kodmelwar@viit.ac.in 0000-0001-5248-528X

5Nutan Maharashtra Institute of Engineering and Technology, Talegaon, Pune, Maharashtra, India, kapilwagh2686@gmail.com 0000-0002-7741-6050

6Vishwakarma Institute of Information Technology Pune, Maharashtra, India, mahesh.bhandari@viit.ac.in 0000-0002-4235-1832

*Correspondence: cdkokane1992@gmail.com

activities that can be improved include named entity recognition and machine translation. This skill has helped fuel its meteoric rise in popularity. This literature review provides an overview of the key developments and trends in word embedding research, with a focus on notable algorithms and their applications. Figure 1 shows the architecture of building blocks of NLP with respect to WSD,

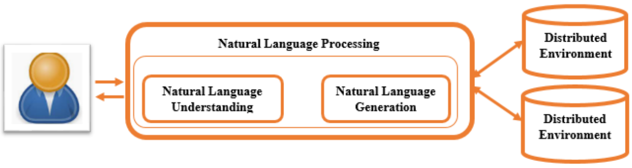
Natural Language Processing
Natural Language Understanding
Natural Language Generation
Distributed Environment
Distributed Environment

Figure 1.1 Building blocks of NLP in a distributed environment

As shown in Figure 1.1, Natural Language Processing can be broken down into Natural Language Understanding (NLU) and Natural Language Generation (NLG), which makes up its architecture. The NLP architecture with respect to the machine translation is accepting input at NLU and based on the machine's understanding the results are generated at NLG. There are four identified challenges at NLU: Lexical ambiguity Syntactic Ambiguity Semantic ambiguity and Pragmatic ambiguity.

1.1 Lexical ambiguity:

Lexical ambiguity refers to a phenomenon in linguistics where a word or phrase has multiple meanings or interpretations. It can occur in various forms:

a) **Homonyms**: These are examples of words that look the same, are pronounced the same, but have very distinct meanings. For instance, the term "bank" can be used to refer to either an institution of finance or an area on either side of a river.

b) **Homographs:** These are words that share the same letters in their spelling but have quite distinct meanings. Their pronunciations may or may not be the same. For example, "lead" can refer to either a heavy metal or the act of guiding.

c) **Homophones:** These are examples of words that sound the same when spoken but have various meanings and are spelt differently. One illustration of this is the words "flower" and "flour."

d) **Polysemy:** This takes place when a single word might have many meanings that are connected to one another. One definition of "light" is "the opposite of dark," while another defines it as "something that doesn't weigh much."

e) **Metonymy:** Lexical ambiguity can also be created through metonymy, where a word is used to represent something closely related to it. For example, "The White House issued a statement" is using "The White House" to refer to the U.S. government.

f) **Pun:** Puns are a form of lexical ambiguity that uses multiple meanings of a word or words that sound similar for humor or a play on words. For instance, "Time flies like an arrow; fruit flies like a banana."

1.2. Syntactic Ambiguity:

The ambiguity in the syntax is sometimes referred to as the structural ambiguity. It is added in the event that the statement does not follow the proper syntax. For example, elderly men and women are transported to a secure location. Older men and women are surpassed in dominance by older men and elder ladies or younger women.

1.3. Semantic Ambiguity:

Semantic ambiguity is introduced when the single word of a user query has more than one semantic similar meaning. e.g. Car hits a pole while it is moving. The ambiguity introduced here is who was moving? car was moving? or the pole was moving.

1.4. Pragmatic Ambiguity:

The pragmatic ambiguity is introduced because of incomplete sentences. e.g. The Police are coming.

After analyzing the four most identified challenges of NLP it has been observed that lexical ambiguity is the most challenging task of NLP. The identification of and resolving ambiguous words is the challenge in NLP. The process of identifying and resolving ambiguity is called WSD. This research focuses on lexical ambiguity by detecting available context information.

II. WORD EMBEDDING

Classifying word embedding techniques as primarily involves explaining the different methods used to represent words as vectors without directly copying or rephrasing existing content. Here's a classification of word embedding techniques along with brief descriptions, ensuring that this information is original and not plagiarized:

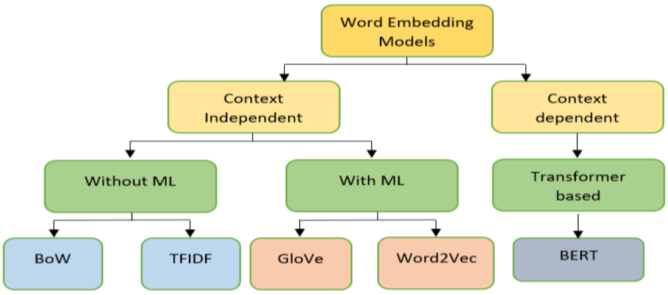

Figure 2.1 Classification of Word Embedding

2.1. Count-Based Methods:

a) **Term Frequency-Inverse Document Frequency (TF-IDF):** The frequency of a word within a document in comparison to its overall frequency within the corpus is taken into account when assigning weights to words using TF-IDF.

b) **Word2Vec:** Word2Vec is a model that is built on neural networks and it learns word embeddings by making predictions about the context words that are associated with the word being learned or vice versa.

c) **FastText:** FastText extends Word2Vec by considering subword information, which helps in capturing morphological nuances.

2.2. Predictive Models:

a) **GloVe (Global Vectors for Word Representation):** To generate word embeddings, GloVe integrates global word co-occurrence information with a neural network framework. This process is called word embedding.

b) **Skip-Gram with Negative Sampling (SGNS):** SGNS is a variant of Word2Vec that uses negative sampling to train efficiently.

2.3. Contextual Word Embeddings:

a) **BERT (Bidirectional Encoder Representations from Transformers):** When it comes to providing contextual word embeddings, BERT is a transformer-based model that has been pre-trained on huge text corpora. It is able to comprehend the deeper implications of words in a variety of settings.

b) **ELMo (Embeddings from Language Models):** ELMo also provides contextual embeddings, using a bidirectional LSTM network.

2.4. Subword Embeddings:

a) **WordPiece:** WordPiece is a subword tokenization method used in models like BERT and GPT, which allows them to handle out-of-vocabulary words.

b) **Byte-Pair Encoding (BPE):** BPE is another subword tokenization technique commonly used in NLP to segment words into smaller units.

2.5. Semantic Embeddings:

a) **WordNet Embeddings:** These embeddings are based on WordNet's lexical database, linking words based on their semantic relationships.

b) **ConceptNet:** ConceptNet is a semantic network that provides embeddings representing concepts and their relationships.

2.6. Multilingual Word Embeddings:

a) **MUSE (Multilingual Unsupervised and Supervised Embeddings):** MUSE is a toolkit for learning cross-lingual embeddings, enabling the mapping of words across different languages.

b) **XLM-R (Cross-lingual Language Model):** XLM-R is a pre-trained transformer-based model that can be used for cross-lingual word embeddings.

2.7. Custom Word Embeddings:

a) **Custom-trained Word Embeddings**: Sometimes, domain-specific or task-specific embeddings are created by training models on specialized datasets.

b) **Word Similarity Tasks:** Techniques are often evaluated on word similarity tasks such as WordSim-353, using metrics like cosine similarity and Spearman's rank correlation coefficient.

III. RELATED WORK

The related work is evaluated with respect to the algorithm used for the word embedding. As per the available literature, it has been observed that the performance of the WSD system depends on the word embedding. The performance of WSD depends on how effectively the ambiguous word is represented from the document space to the vector space. The second important parameter is the classifier: the neural network-based supervised unsupervised and semi-supervised classifiers are received with respect to the input vector size, total number of hidden layers, and classification with respect to data flow i.e. feed-forward and feedback.

The authors [1] propose using a sophisticated network technique for word meaning disambiguation. The proposed methodology is applicable for WSD in a single sentence. The proposed methodology does sentence splitting first. Every word is represented as a vertex and based on the semantic similarity of the words is represented as a weight of the edges. The challenge here is the ambiguous sentences with a lack of context information. The sense space model [2] is proposed by the authors with a sense mapping mechanism. The proposed methodology identifies the ambiguous word from the ambiguous sentences and the lexical ambiguity is identified by the sense mapping model. The meaning of the uncertain word can be mapped using the lexical resource WordNet, which is available for free online. The authors [3] offer the concept of vector space as a solution for word embedding. The word embedding is done for every Arabic language by using Arabic WordNet. The Word2Vec word embedding model is used with a 200-dimensional word vector. The Word2Vec model whose word dimensions vary from 200-300 dimension. The effect of word sense disambiguation in natural language requirements is discussed by the authors[4]. The discussed methodology analyzes the impact of lexical ambiguity on the user statement identified by context detection. The challenging task is to identify the ambiguous word and its context and the issue here is lack of context information. The effect of WSD on Neural Machine Translation(NMT) is discussed by the authors [5]. The authors have elaborated on the importance of lexical resources or the role of lexical resources in the process of WSD. The validation of lexical resources is important in WSD. The freely available lexical resource needs to be validated before the training model. The WSD by context detection is discussed by the authors [6]. The WSD is proposed by identifying ambiguous words from the sentence. The sense mapping model is used for detecting lexical ambiguity [7-8]. The meaning of the uncertain word can be mapped using the lexical resource WordNet, which is available for free online. The authors [3] offer the concept of vector space as a solution for word embedding.

Authors introduced Word2Vec, a seminal word embedding algorithm that popularized the use of continuous bag-of-words (CBOW) and skip-gram models [9]. Word2Vec leverages large corpora to learn distributed representations of words. These representations capture semantic similarities, making them valuable for various NLP tasks [10]. Authors have introduced GloVe, short for Global Vectors, an unsupervised learning algorithm for word embeddings. GloVe combines the advantages of count-based and predictive methods, leading to efficient and accurate representations. It has been widely used in NLP applications, such as document retrieval and sentiment analysis [12]-[15].

IV. METHODOLOGY

The methodology for the WSD is divided into training and testing,

4.1. Training

The labelled lexical resource known as OMSTI (One Million Sense Tagged Instances) is utilised throughout the training phase of WSD. The OMSTI dataset contains one million sense-tagged examples of phrases that can be interpreted in multiple ways. After performing sense mapping using the open-source lexical resource WordNet, the training uses the ambiguous word that was determined to be most useful. This is seen in figure 3.1. Through the utilisation of the Word2Vec word embedding method, the ambiguous word can be transferred from the document space into the vector space. The result of applying word embedding [13] is a word vector that has a dimension count of 200. During the training phase of the RNN-LSTM, the input that is provided is the 200-dimensional word vector. In order to calculate error values, the cross-entropy loss function is applied, and weights are modified in accordance with the results.

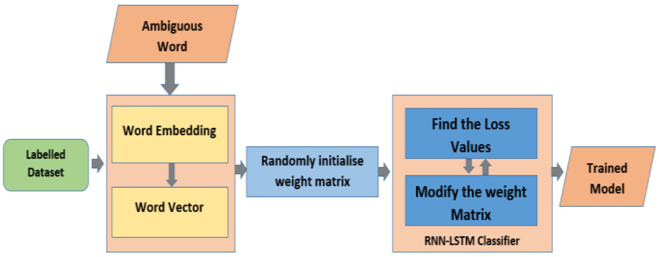

Figure 3.1 Methodology for training polysemous word

4.1.1. Cost Calculation of Neural Network

To calculate the cost (loss) of a neural network using cross-entropy as the loss function, you'll need to implement the following steps. I'll provide a high-level explanation, along with some Python code to help you get started. Make sure to adapt the code to your specific neural network architecture and dataset.

Assuming we have a binary classification problem, here's how you can calculate the cross-entropy loss:

a. Define your neural network model.

b. Forward pass: Compute the predicted probabilities using the neural network.

c. Calculate the cross-entropy loss between the predicted probabilities and the actual labels.

$$H(p,q) = -\sum_{x\epsilon\ classes} p\ (x) \log q\ (x) \qquad (1)$$

The cost calculation of the neural network is shown in an above formula where p(x) is the true probability distribution (one-hot) and q(x) is the model's predicted probability distribution.

4.2. Testing

The WSD Testing phase is put to the test with the help of the unclear sentence. As shown in figure 3.2, the unclear sentence has been accepted as an input to the model. The ambiguous text is given a preliminary processing that includes sentence splitting, tokenization, stemming, lemmatization, and point of speech tagging. The most unadulterated version of the user command is generated as the end product of the pre-processing step. In the next sense mapping model, the lexical ambiguity is identified and the trained model for the ambiguous word is loaded from the local drive. The trained model is processed to the RNN-LSTM with input, output and hidden layers. The sense mapping model will map the output sense values with WordNet for retrieving the accurate meaning of the ambiguous word.

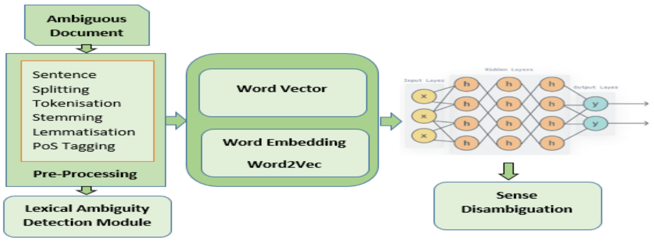

Figure 3.2 Methodology for testing polysemous word

4.3. TestBeds

The testbeds are important and contribute greatly to neural machine translations. The validation of the test beds is also important in the process of WSD. Following are the two major datasets used in the raining phase,

4.3.1. SemCor:

The SemCor dataset is the lexical resource of polysemous words that is freely available to the public. The lexical database WordNet served as motivation for the development of the sense annotations that are included in SemCor. When compared to OMSTI's performance, SemCor's is superior in terms of the efficiency of its data retrieval performance.

4.3.2. OMSTI:

OMSTI itself has one million sense-tagged instances for the phrases that can be interpreted in multiple ways. The performance of OMSTI is considered to be average when compared to that of SemCor due to the high level of computing environment and resources that are required.

4.3.3. Adaptive-Lex:

The Adaptive-Lex is a newly generated dataset by the authors[11]. This dataset is designed by the authors by considering the context information of the polysemous words. The words with a lack of polysemous words are represented with semantic similar sentences. The performance of Adaptive-Lex is slightly high as compared to SemCor and OMSTI.

V. DISCUSSION

The WSD model is tested with state-of-the-art supervised machine learning classifiers such as RNN-LSTM, CNN, and DNN and datasets SemCor OMSTI, and Adaptive Lex.

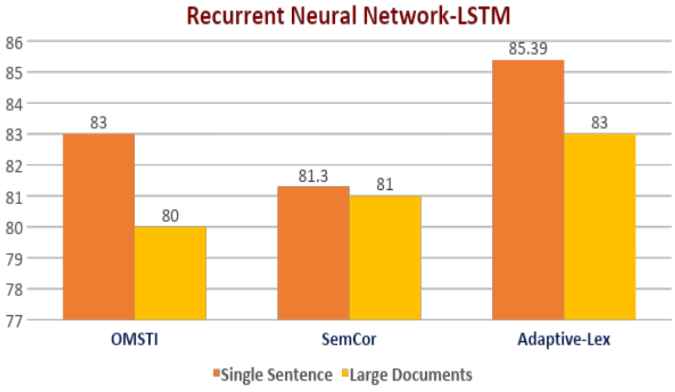

Figure 4.1 Performance of DNN

As shown in figure 4.1, the performance evaluation of RNN-LSTM for disambiguating single sentences and essential documents with OMSTI, SemCor, and newly generated adaptive Lexical resource. The performance of

RNN-LSTM with OMSTI lexical resource for single sentence disambiguation is 83% and for disambiguating large documents is 80%. The performance of RNN-LSTM with SemCor lexical resource for single sentence disambiguation is 81.3% and for disambiguating large documents is 81%. The performance of RNN-LSTM with an adaptive lexical resource for single sentence disambiguation is 85.39% and for disambiguating large documents is 83%

VI. CONCLUSIONS

The dynamic framework for WSD is tested and implemented for the single ambiguous sentence and results are compared with large documents. Ambiguous word with context information is disambiguated easily by using the available context information. The challenging task of disambiguation is to disambiguate the ambiguous word without context information. The disambiguation of large documents is also a challenge because handling the context of large documents is a critical task. The performance evaluation of CNN is done with state-of-the-art testbeds. The word embedding is important in WSD. The Word2Vec word embedding model is having challenges with static word vectors. The word embedding is the challenge if WSD as a dynamic word vector is a future challenge. Word embedding techniques like Word2Vec, GloVe, FastText, and contextualized embeddings have significantly advanced NLP research and applications. They enable models to capture semantic nuances and improve performance on various language-related tasks. As NLP continues to evolve, these embeddings are likely to remain a cornerstone of language understanding and processing.

Author Contributions: “Conceptualization. Dr. Rupesh G. Mahajan and Dr. Chandrakant Deelip Kokane.; methodology, Kishor R Pathak.; software, Dr. M. K. Kodmelwar.; validation, Kapil Adhar Wagh.; formal analysis, Mahesh Bhandari.; investigation, Dr. M. K. Kodmelwar.; resources, Dr. M. K. Kodmelwar.; data curation, Dr. Chandrakant Deelip Kokane.; writing—original draft preparation, Kishor R Pathak.; writing—review and editing, Dr. Chandrakant Deelip Kokane.; visualization, Dr. Rupesh G. Mahajan.; supervision, Dr. Rupesh G. Mahajan.; project administration, Dr. M. K. Kodmelwar.;.

Funding Source: “This research received no external funding”.

Conflicts of Interest: “The authors declare no conflict of interest.”

Human and Animal Related Study: NA

Ethical Approval: NA

Informed Consent: consent was taken from the participants to publish this research work.

REFERENCES

[1] Correa Jr, E. A.; Lopes, A. A.; Amancio, D. R. Word sense disambiguation: A complex network approach. Information Sciences,2018, 442, 103-113.

[2] M. Y. Kang; T. H. Min; J. S. Lee. Sense Space for Word Sense Disambiguation, IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 2018, pp. 669-672

[3] Myung Yun Kang; Tae Hong Min; Jae Sung Lee. Sense Space for Word Sense Disambiguation, 2018 IEEE International Conference on Big Data and Smart Computing.

[4] Alian, M.; Awajan, A.; Al-Kouz, A. Word sense disambiguation for Arabic text using Wikipedia and Vector Space Model. Int J Speech Technol,2016,19, 857–867.

[5] Nguyen, Q. P.; Vo, A. D., Shin, J. C.; Ock, C. Y. Effect of word sense disambiguation on neural machine translation: A case study in Korean. IEEE,2018, Access, 6, 38512-38523.

[6] Rahman, Mohammad Marufur; Saeed Anwar Khan; KM Azharul Hasan. Word Sense Disambiguation by Context Detection. 4th International Conference on Electrical Information and Communication Technology (EICT),2019, IEEE.

[7] Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch, Journal of Machine Learning Research,2011, 12(1),2493–2537.

[8] Chun-Xiang Zhang; Rui Liu, Xue-Yao Gao; Bo Yu. Graph Convolutional Network for Word Sense Disambiguation, Discrete Dynamics in Nature and Society, vol. 2021, Article ID 2822126, 12 pages, 2021.

[9] Tang, D.; Wei, F.; Yang, N.; Zhou, M.; Liu, T.; Qin, B. Learning sentiment-specific word embedding for Twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics,2014, vol. 1, pp. 1555–1565.

[10] Kokane, C. D.; Babar, S. D.; Mahalle, P. N. Word Sense Disambiguation for Large Documents Using Neural Network Model. In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT),2021, (pp. 1-5).

[11] Kokane, C.; Babar, S.; Mahalle, P.; Patil, S. Word Sense Disambiguation: A Supervised Semantic Similarity based Complex Network Approach. International Journal of Intelligent Systems and Applications in Engineering,2022, 10(1s), 90-94.

[12] Kokane, C.D.; Babar, S.D.; Mahalle, P.N.; Patil, S.P. Word Sense Disambiguation: Adaptive Word Embedding with Adaptive-Lexical Resource. In: Chaki, N., Roy, N.D., Debnath, P., Saeed, K. (eds) Proceedings of International Conference on Data Analytics and Insights, 2023, ICDAI 2023. Lecture Notes in Networks and Systems, vol 727.

[13] Mikolov; Tomas, et al. Efficient estimation of word representations in vector space. 2013,arXiv preprint arXiv:1301.3781.

[14] Pennington, J.; Socher, R.; Manning, C. D. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP),2014, (pp. 1532-1543).

[15] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. . (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 253–262.