# A comparative study of Word Sense Disambiguation in Natural Language Processing using Sentiment Analysis and Deep Learning models

Malavika R[1] Shribala S[2] Pranav Kannan[3] Sathyan S[4]

## Abstract:

Word Sense Disambiguation (WSD) is a fundamental task in natural language processing that aims to identify the appropriate sense of a word in a given context. In the realm of sentiment analysis, accurate WSD is crucial for determining the sentiment expressed in text using predefined sentiment lexicons. This research paper presents a comparative study of two popular approaches, one using Bidirectional Long Short-Term Memory (Bi-LSTM) and Bidirectional Gated Recurrent Unit (Bi-GRU), and the other using Lexicon based sentiment Analysis for word sense disambiguation.

The lexicon-based approach begins by constructing a dataset by scraping data from various sources, which then utilizes sentiment lexicons to assign sentiment polarity. The deep learning approach, which employs Bi-LSTM and Bi-GRU models to capture complex contextual dependencies for word sense disambiguation uses WiC(Word-in-Context) for the evaluation of context-sensitive word embeddings. WiC is framed as a binary classification task. Each instance in WiC has a target word w, either a verb or a noun, for which two contexts are provided.

Evaluation metrics such as accuracy, precision, recall, and F1-score are used to compare the results. The findings reveal that the deep learning models, Bi-LSTM and Bi-GRU, outperform the lexicon-based approach in terms of overall accuracy. The ability of these models to capture intricate contextual relationships and learn effective representations contributes to their superior performance.

This research contributes to the understanding of word sense disambiguation in sentiment analysis and provides insights into the effectiveness of deep learning models and lexicon. The results highlight the potential of these models for improving sentiment analysis systems by accurately disambiguating word senses and enhancing sentiment classification accuracy.

## 1. Introduction:

Word Sense Disambiguation (WSD) is a fundamental task in natural language processing (NLP) that aims to determine the correct meaning of a word in a given context. It is essential to many NLP applications, such as text summarization, information retrieval, machine translation, and question-answering. Word ambiguity results from the fact that, depending on the context in which it is used, a single word may have several meanings, also referred to as word senses. By assigning the most suitable sense to each occurrence of a term, WSD eliminates this uncertainty.

WSD is required because human language is inherently complicated. Depending on the term's specialized domain, the words around it, and the sentence's general context, a word may have a variety of interpretations. For instance, the term "bank" can be used to describe both a financial organization and the river's margin. For accurate comprehension and interpretation of a text, determining the appropriate sense is essential.

WSD systems must be evaluated to determine how well they are working. To compare various WSD methods, several common evaluation datasets, including SemCor, Senseval, and SemEval, have been created. Accuracy and F1 score are two evaluation metrics that are used to gauge how well these systems function. Even though WSD research has advanced significantly, it is still a difficult undertaking, particularly when there are numerous senses of a word that are closely connected or when the context is very ambiguous. Additionally, language-specific phenomena like idioms and cultural allusions, which call for a deeper linguistic and cultural understanding, can have an impact on WSD.

To discover the correct meaning of words in various circumstances, word sense disambiguation is an important NLP task. For many NLP applications, it has real-world ramifications. To address this issue, researchers have created a variety of strategies, including knowledge-based, supervised, unsupervised, and neural network-based methods. Research is still being done in the areas of assessing the effectiveness of WSD systems and resolving any outstanding issues.

## 2. Literature Review:

In his extensive survey, Navigli (2009) offers a thorough examination of Word Sense Disambiguation (WSD). The survey aims to provide a comprehensive understanding of the approaches, techniques, and evaluation methods utilized in WSD research. Navigli covers various facets of WSD, including knowledge-based methods, supervised and unsupervised learning approaches, and hybrid systems that integrate multiple techniques. The survey addresses the challenges and limitations associated with WSD, such as sparse data, lexical and semantic ambiguity, and the absence of extensively annotated corpora. Additionally, Navigli underscores the significance of evaluation standards and datasets for benchmarking WSD systems. The survey concludes by summarizing the major trends and future directions in WSD research, emphasizing the necessity for robust algorithms capable of handling multilingual, domain-specific, and context-dependent disambiguation tasks. Overall, the survey serves as an invaluable resource for researchers and practitioners in the field of WSD, offering a comprehensive overview of the cutting-edge advancements at the time. [11]

In their contribution to "The Oxford Handbook of Computational Linguistics," Stevenson and Wilks (2003) provide a comprehensive exploration of Word Sense Disambiguation (WSD). The authors highlight the significance of WSD in the field of natural language processing and address the challenges associated with it. They discuss various approaches utilized in WSD, such as knowledge-based methods, supervised and unsupervised learning, and corpus-based approaches. Additionally, the authors examine the role of lexical resources and computational lexicons in WSD and emphasize the importance of evaluation methodologies for assessing WSD system performance. Throughout the chapter, Stevenson and Wilks stress the continuous nature of WSD research and advocate for ongoing efforts in developing robust and accurate disambiguation techniques. [12]

### 2.1. Word Sense Disambiguation using Sentiment Analysis:

In the paper titled "Sentiment analysis of figurative language using a word sense disambiguation approach," Rentoumi et al. present a novel approach to sentiment analysis that incorporates word sense disambiguation (WSD) techniques. The authors highlight the importance of effectively analyzing sentiment in figurative language and propose a method that combines WSD with sentiment analysis to address the challenges associated with figurative expressions.

The study demonstrates that the integration of WSD significantly enhances the accuracy of sentiment analysis in the context of figurative language. By disambiguating the senses of words, the proposed approach enables a more precise understanding of their intended meanings within figurative expressions. This improved understanding of figurative language contributes to a more accurate sentiment analysis of such texts.

The findings of the study suggest that WSD techniques hold great potential for enhancing sentiment analysis in texts containing figurative language. The successful integration of WSD in sentiment analysis provides a valuable tool for researchers and practitioners interested in analyzing sentiment in contexts where figurative expressions are prevalent. [1]

In the paper "Word sense disambiguation based sentiment lexicons for sentiment Classification" by Hung and Chen (2016), the authors demonstrate that word sense disambiguation (WSD) can enhance the precision of sentiment classification. Their proposed method employs sentiment lexicons to clarify the different meanings of words within a sentence. Through an evaluation using a movie review dataset, the authors establish that their approach leads to a potential improvement of sentiment classification accuracy by up to 10%.

The authors highlight the significance of WSD in sentiment classification, emphasizing its ability to resolve ambiguities arising from words with multiple senses. Their method for WSD involves utilizing sentiment lexicons, which consist of manually labelled words indicating their sentiment polarity (positive, negative, or neutral). By employing sentiment lexicons, the authors disambiguate word senses by identifying the sense that aligns most consistently with the overall sentiment polarity of the sentence.

To evaluate the effectiveness of their WSD method, the authors conduct experiments on a dataset comprising movie reviews. The results reveal a noteworthy enhancement in sentiment classification accuracy, supporting the authors' claim that WSD can serve as a valuable technique for improving the precision of sentiment classification. [2]

In their study on "Word sense disambiguation for lexicon-based sentiment analysis," Pamungkas and Putri (2017) emphasize the potential of word sense disambiguation (WSD) in enhancing the precision of lexicon-based sentiment analysis. The researchers propose a novel approach to WSD employing a graph-based Lesk algorithm, which they subsequently evaluate using a dataset comprised of movie reviews. Through their experiments, they demonstrate that their method contributes to an improvement of up to 5% in the accuracy of lexicon-based sentiment analysis.

The authors contend that WSD plays a crucial role in lexicon-based sentiment analysis due to its capacity to resolve word meanings that possess multiple senses. To address this challenge, they introduce the graph-based Lesk algorithm as their method for WSD. This algorithm operates by constructing a graph encompassing all conceivable senses of a given word and then traversing this graph to determine the sense most closely aligned with the sentence's context.

To validate the efficacy of their proposed approach, the authors conduct an evaluation using a dataset consisting of movie reviews. Their findings affirm that the implementation of their method yields noticeable enhancements, up to a 5% increase in accuracy, within lexicon-based sentiment analysis. Consequently, the results presented in the paper indicate that WSD can be an invaluable technique for enhancing the precision of lexicon-based sentiment analysis. [3]

In their paper "Prediction of election results by enhanced sentiment analysis on Twitter Data using Word Sense Disambiguation," Jose and Chooralil (2015) present their findings on the effectiveness of word sense disambiguation (WSD) in enhancing sentiment analysis on Twitter data. Through their research, they propose a novel approach to WSD utilizing a graph-based Lesk algorithm. By applying their method to a dataset comprising tweets from the 2014 Indian general election, they demonstrate that their technique can significantly enhance the accuracy of sentiment analysis by up to 10%.

The authors emphasize the importance of WSD in sentiment analysis on Twitter data, as it aids in disambiguating words that possess multiple senses. Their approach to WSD involves the utilization of a graph-based Lesk algorithm, which initially constructs a graph representing all possible senses of the word. The algorithm then traverses the graph to identify the sense of the word that best aligns with the tweet's context.

To evaluate the effectiveness of their proposed WSD method, the authors conduct experiments using a dataset of tweets from the 2014 Indian general election. The results demonstrate that their approach leads to a significant improvement of up to 10% in sentiment analysis accuracy. Based on these findings, the authors suggest that WSD can be a valuable technique for enhancing the accuracy of sentiment analysis on Twitter data. [4]

In the research paper by Seifollahi and Shajari (2019), the authors explore the use of Word Sense Disambiguation (WSD) in sentiment analysis of news headlines for predicting the FOREX market. Their goal is to improve the accuracy of sentiment analysis by clarifying the meanings of words found in news headlines. To achieve this, the authors propose a practical approach that combines WSD techniques with sentiment analysis algorithms.

The study involves conducting experiments and evaluations to assess the performance of sentiment classification models when WSD is incorporated. The results indicate that the inclusion of WSD in sentiment analysis significantly enhances the effectiveness of sentiment classification models within the context of FOREX market prediction. By disambiguating the various senses of words, the proposed approach enables a more precise interpretation of the sentiment expressed in news headlines. Consequently, this improvement in sentiment analysis accuracy contributes to better predictions of market trends. [5]

### 2.2. Word Sense Disambiguation using Deep Learning Models:

In their research paper, Calvo et al. (2019) address the challenge of Word Sense Disambiguation (WSD) by proposing a universal solution using deep neural networks. By incorporating word and context embeddings, the authors develop a neural network architecture capable of capturing semantic information for disambiguation. Through extensive experimentation, they demonstrate that their deep neural network-based approach outperforms existing methods. The proposed system exhibits high accuracy in disambiguating word senses across various languages and domains, suggesting its potential for universal applicability. These findings indicate that deep neural networks effectively leverage contextual information and word embeddings, leading to improved accuracy and versatility in WSD, thereby advancing the field of natural language processing tasks. [6]

In their research, Nithyanandan and Raseek (2019) compare different deep learning models to address the challenge of Word Sense Disambiguation (WSD). The authors investigate the performance of Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and Transformer models in disambiguating word senses. By conducting experiments and analyzing the results, they find that deep learning models, particularly the Transformer architecture, outperform traditional approaches in WSD. These findings underscore the significance of utilizing deep learning techniques in WSD, as they effectively capture contextual information and semantic relationships, thereby enhancing the accuracy and precision of word sense disambiguation systems. Ultimately, this study contributes to the advancement of understanding and application of deep learning models for WSD tasks. [7]

In the study conducted by Saidi and Jarray (2022), the authors explore the field of Arabic Word Sense Disambiguation (WSD) by incorporating BERT representations and a Part-of-Speech (POS) tagger. Their primary objective is to enhance the accuracy of WSD in Arabic by utilising the contextualized embeddings provided by BERT and the linguistic information from POS tags. To achieve this, the authors propose a novel methodology that combines BERT representations and POS tagging to capture both syntactic and semantic features for disambiguating word senses.

Through a series of experiments and evaluations, the researchers establish that the integration of BERT representations and POS tagging leads to substantial improvements in Arabic WSD when compared to using either approach independently. This conclusion emphasizes the effectiveness of incorporating both contextualized embeddings and linguistic information in enhancing the disambiguation of word senses in Arabic. Consequently, this research contributes to the advancement of WSD techniques for analyzing Arabic text. [8]

Ahmed, Samee, and Mercer (2018) present a new neural sequence model in their paper, aiming to improve Word Sense Disambiguation (WSD). By introducing multiple attention mechanisms into the neural network architecture, the authors enhance the accuracy of WSD. The proposed model incorporates attention mechanisms at various levels, including word level, context level, and attention over different senses. This multi-attention approach enables the model to effectively capture and weigh relevant contextual information for word sense disambiguation. The authors conducted experiments and evaluations to compare their model against existing WSD methods, and the results demonstrated that their proposed model achieves superior accuracy. These findings suggest that integrating multiple attention mechanisms in a neural sequence model can greatly enhance the performance of WSD, leading to more precise and context-aware disambiguation of word senses. [9]

Wang, Zhang, Yu, and Zhang (2022) propose a method for generating Contextual Sentiment Embeddings (CSE) using a bi-directional Gated Recurrent Unit (GRU) language model. The study focuses on capturing contextual information in sentiment analysis by representing text in a sentiment-aware embedding space. The authors introduce a framework that utilizes bi-directional GRU to model context and generates sentiment-aware embeddings.

By training the model on a large dataset, they obtain embeddings that effectively encode sentiment information from the surrounding text. Extensive experiments and evaluations demonstrate the superiority of their approach in sentiment analysis, surpassing existing methods in accuracy. These findings underscore the significance of contextual information in sentiment analysis and highlight the potential of bi-directional GRU language models in generating informative and sentiment-aware embeddings. This research contributes to advancing the understanding and application of contextual sentiment representations in various knowledge-based systems. [10]

## 3. Architecture:

### 3.1. Long Short-Term Memory - LSTM:

LSTM stands for Long Short-Term Memory, and it is a type of recurrent neural network (RNN). It is widely used where information from earlier time steps needs to be carried forward and used to make predictions at later time steps. The key difference between LSTM and traditional RNN is the presence of gates. There are three types of gates in an LSTM network forget gate, input gate and output gate: [14]

The forget gate (1) decides which information to discard from the previous hidden state and the value of it can be represented as,

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \textbf{ (1)}$$

The input gate (2) determines which new information to be added to the current hidden state and the value of it can be represented as

$$i_t = \sigma (W_i [h_{t-1}, x_t] + b_i) \textbf{ (2)}$$

The output gate (3) decides which information to be given as output from the current hidden state and the value of it can represented as

$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o) \textbf{ (3)}$$

In the equations, 1,2 and 3 the $b_x$ and $W_x$ values represent the bias and weight values of respective gates. The output of the previous timestep and input of the current timestep of each gate is denoted as $h_{t-1}, x_t$ respectively.

LSTM calculates the value of candidate state (4) which is represented $\hat{C}_t$ as to get the value of cell state $C_t$.

$$\hat{C}_t = tanh(W_i [h_{t-1}, x_t] + b_c \textbf{ (4)}$$

The cell state (5) which is responsible for the memory at a particular timestamp can be denoted as

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad \textbf{(5)}$$

The value of the final output (6) of LSTM is calculated using the equation,

$$h_t = o_t * tanh(C_t) \quad \textbf{(6)}$$

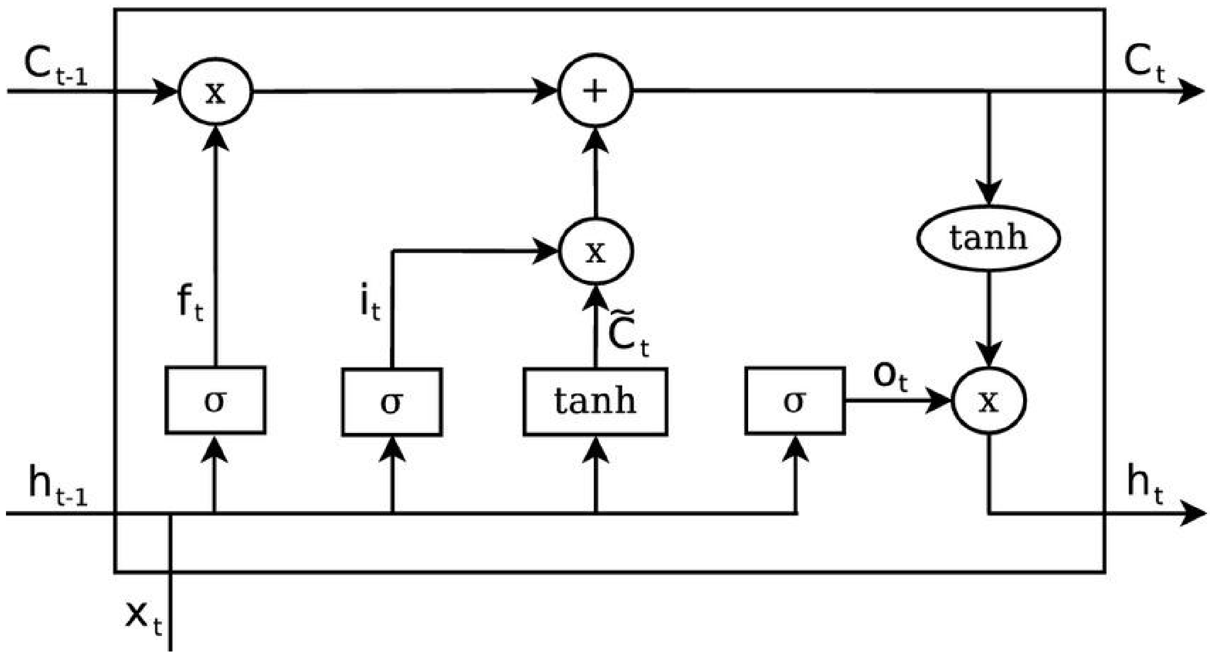


*Figure 3.1: Illustration of LSTM Architecture [15]*

### 3.2. Gated Recurrent Units - GRU:

GRU stands for Gated Recurrent Unit, and it is another type of recurrent neural network (RNN) that is similar to LSTM but with a simplified architecture. The key difference between GRU and LSTMs is that GRU have only two gates, whereas LSTMs have three gates. The two gates in a GRU network are reset and update gates [14]

The reset gate (7) determines how much of the previous hidden state to forget and it can be represented as

$$r_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad \textbf{(7)}$$

The update gate (8) determines how much of the new information to add to the current hidden state and it can be represented as

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right) \quad \textbf{(8)}$$

The intermediate values (9)(10) obtained from the update and reset gates are denoted as $\hat{h}_t$ and $h_t$ respectively and the equation of both those values are

$$\hat{h}_t = tanh(W\,[r_t * h_{t-1}, x_t])\ \textbf{(9)}$$

$$h_t = \left(1 - z_t\right) * h_{t-1} + z_t * h_t\ \textbf{(10)}$$

Hyperbolic tangent function is represented as tanh, and the sigmoid function is denoted as σ.

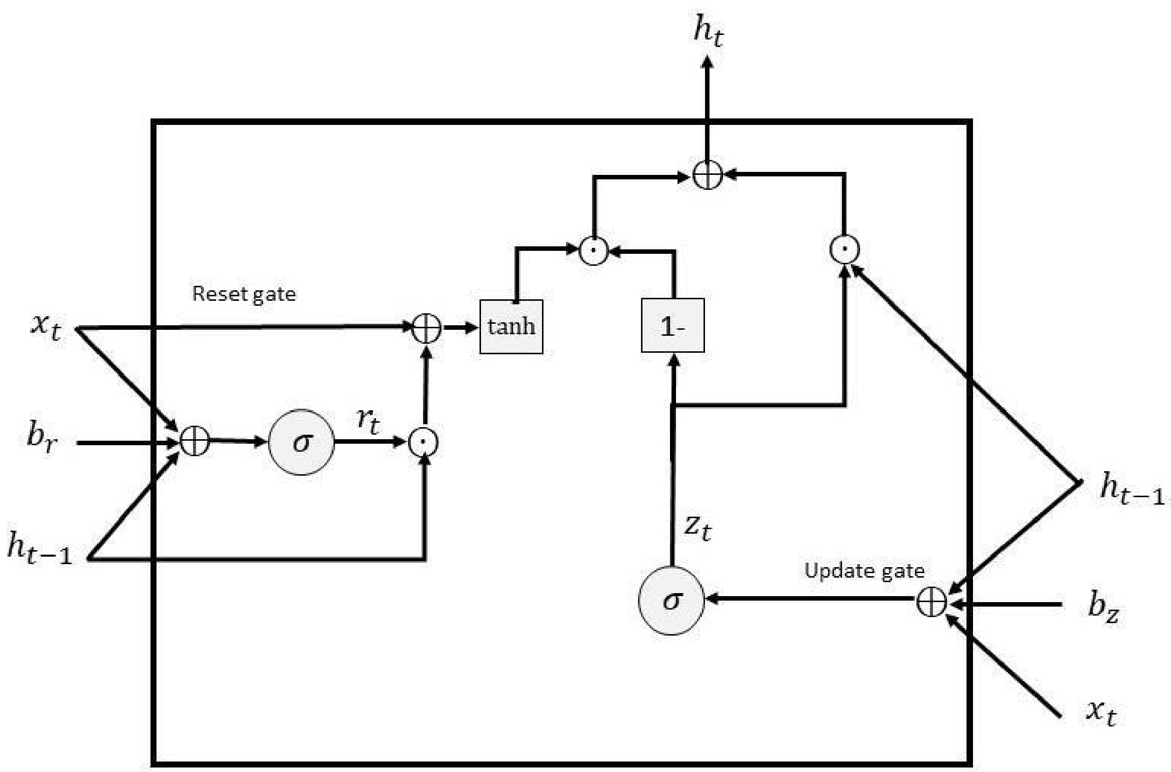


*Figure 3.2: Illustration of GRU architecture [16]*

## 4. Experimental Setup:

For the comparative study, we have taken two approaches to perform Word Sense Disambiguation on two different methods. One approach uses sentimental analysis to perform WSD on a Hindi language dataset while the other approach uses Deep Learning models like Bi-LSTM and Bi-GRU to perform WSD on a WiC (word-in-context) dataset.

### 4.1. Implementation using Lesk Approach:

The experimental setup begins with the development of a dataset containing sentences in Hindi, collected from various sources, to conduct the experiments.

To facilitate the experiments, necessary libraries are imported, and NLTK resources are downloaded using the nltk.download() function. This ensures access to tokenizers and stopwords required for preprocessing.

A preprocess() function is implemented to preprocess the input text. It tokenizes the text using NLTK's word_tokenize() function, converts the tokens to lowercase, removes non-alphabetic words, and filters out English stopwords using NLTK's stopwords corpus.

For sentiment analysis on the Hindi text, a translation to English is performed using the deep translator library. Then, the lesk algorithm is applied.

To retrieve the possible word senses for a given word, the get_word_senses() function is implemented. It utilizes WordNet and NLTK's wn.synsets() function, returning a set of synset names representing the different senses of the word.

The sentiment score for a specific sense is calculated using the compute_sentiment_score() function. It accesses SentiWordNet and retrieves the positive and negative scores for the sense. The sentiment score is obtained by computing the difference between these scores.

The get_best_sense() function selects the best sense for a word within a given context. It iterates over the senses of the word, calculates the sentiment score for each sense, and compares them to determine the best sense. The score is then multiplied by the number of overlapping words between the sense's signature (generated using the get_signature() function) and the context words.

The get_signature() function generates the signature of a sense based on its definition and examples. It retrieves the definition and examples of the sense from WordNet, preprocesses them using the preprocess() function, and returns a set of words representing the sense's signature.

To compute the overall sentiment score of a given text, the compute_sentiment() function is employed. It preprocesses the text using the preprocess() function, iterates over the words, and retrieves the best sense for each word using the get_best_sense() function. The sentiment scores of all words are accumulated in the scores list.

The proposed system's output is evaluated to assess its efficiency using four parameters: Accuracy, Precision, Recall, and Fscore. These metrics provide insights into the performance of the system.

### 4.2. Implementation using Bi-LSTM and Bi-GRU:

The dataset [13] consists of labeled instances (with columns representing different attributes, namely, the target word (Column1), context 1 (Column7), context 2 (Column8), and label (Column9). The data is loaded into a DataFrame for further processing.

| | Column1 | Column2 | Column7 | Column8 | Column9 |
|---|---|---|---|---|---|
| **0** | approach | V | Approach a task. | To approach the city. | 0 |
| **1** | summer | V | We like to summer in the Mediterranean. | We summered in Kashmir. | 1 |
| **2** | meet | V | The company agrees to meet the cost of any rep... | This proposal meets my requirements. | 1 |
| **3** | development | N | The organism has reached a crucial stage in it... | Our news team brings you the latest developments. | 0 |
| **4** | narrowness | N | The problem with achievement tests is the narr... | Frustrated by the narrowness of people's horiz... | 1 |
| **...** | ... | ... | ... | ... | ... |
| **5423** | round | N | He ordered a second round. | They brought us a round of drinks about every ... | 1 |
| **5424** | run | N | The team enjoyed a brief run of victories. | Yesterday we did a run of 12,000 units. | 0 |
| **5425** | charge | V | Can I charge this purchase? | The suspect was charged with murdering his wife. | 0 |
| **5426** | catch | V | Catch one's breath. | I caught the hem of my dress in the brambles. | 0 |
| **5427** | rest | N | Now that we're all in agreement, we can put th... | The ocean was finally at rest. | 1 |

*Table 4.1: WiC Dataset for English Language [13]*

To prepare the data for training, the word, context 1, context 2, and label columns are extracted from the DataFrame. Tokenization is performed using a tokenizer, which is fitted on the concatenated context 1 and context 2 texts. The resulting word index is obtained.

The context 1 and context 2 texts are then converted into sequences using the tokenizer. To ensure uniform input size, padding is applied to the sequences, limiting them to a maximum length of 30. The input data is constructed by concatenating the padded context 1 and context 2 sequences. The labels are transformed into one-hot encoded vectors.

For model evaluation, the input data and labels are split into training and validation sets using a 33% test size ratio. This allows for assessing the model's performance on unseen data.

The BiLSTM model is constructed using the Keras Sequential API. The model architecture consists of an embedding layer, a bidirectional LSTM layer, and a dense layer with softmax activation. It is compiled with the Adam optimizer and employs categorical cross-entropy loss as the objective function.

To train the model, the training data is fed into the model and validated using the validation data. The training is conducted over 10 epochs with a batch size of 32.

Following model training, the trained BiLSTM model is evaluated on the validation data to calculate the loss and accuracy metrics.

To optimize the model's hyperparameters, a grid search is performed using the GridSearchCV functionality from scikit-learn. The parameter grid includes different values for the embedding dimension and LSTM units. A function is defined to create the model based on the given hyperparameters. The Keras classifier wrapper is employed to enable compatibility with scikit-learn's grid search. The grid search is conducted using 3-fold cross-validation.

Finally, the best performing hyperparameters, along with the corresponding accuracy, are reported.

This experimental setup allows for the training, evaluation, and hyperparameter optimization of the BiLSTM model for word sense disambiguation. The approach provides a comprehensive methodology for effectively addressing the research problem.

## 5. Results:

The above implementations give us effective approaches for word sense disambiguation for natural languages.

### 5.1. Results using Lesk Approach:

Based on the experimental results, the sentiment analysis model achieved an overall accuracy of 64% in classifying the sentiment of the given data. The precision scores for the "Neutral," "Positive," and "Negative" classes were 50%, 63%, and 80%, respectively. The recall scores for the classes were 25%, 100%, and 27% respectively. The F1-scores for the "Neutral," "Positive," and "Negative" classes were 33%, 77%, and 40%, respectively. The macro-averaged F1-score across all classes was 50%. The weighted average F1-score was 58%. These results indicate that the model performed relatively well in classifying positive sentiments, but struggled with neutral and negative sentiments.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Neutral | 0.50 | 0.25 | 0.33 | 8 |
| Positive | 0.63 | 1.00 | 0.77 | 24 |
| negative | 0.80 | 0.27 | 0.40 | 15 |
| accuracy | | | 0.64 | 47 |
| macro avg | 0.64 | 0.51 | 0.50 | 47 |
| weighted avg | 0.66 | 0.64 | 0.58 | 47 |

*Table 5.1 : Table representing the different metrics using sentiment analysis*

### 5.2. Results using Bi-LSTM and Bi-GRU:

The dataset consists of labeled instances, and the model is trained and evaluated on the data. The Bi-LSTM model with specific hyperparameters achieves notable accuracy in disambiguating word senses. The proposed approach demonstrates the effectiveness of the Bi-LSTM and Bi-GRU architecture in resolving word sense ambiguity. The experimental results highlight the potential of this method for improving natural language processing tasks.

| | Language | BiLSTM Accuracy | BiGRU Accuracy |
|---|---|---|---|
| **0** | English | 0.577009 | 0.573103 |
| **1** | French | 0.583308 | 0.572633 |
| **2** | German | 0.651382 | 0.661572 |

*Table 5.2: Table representing the accuracies of Bi-LSTM and Bi-GRU*

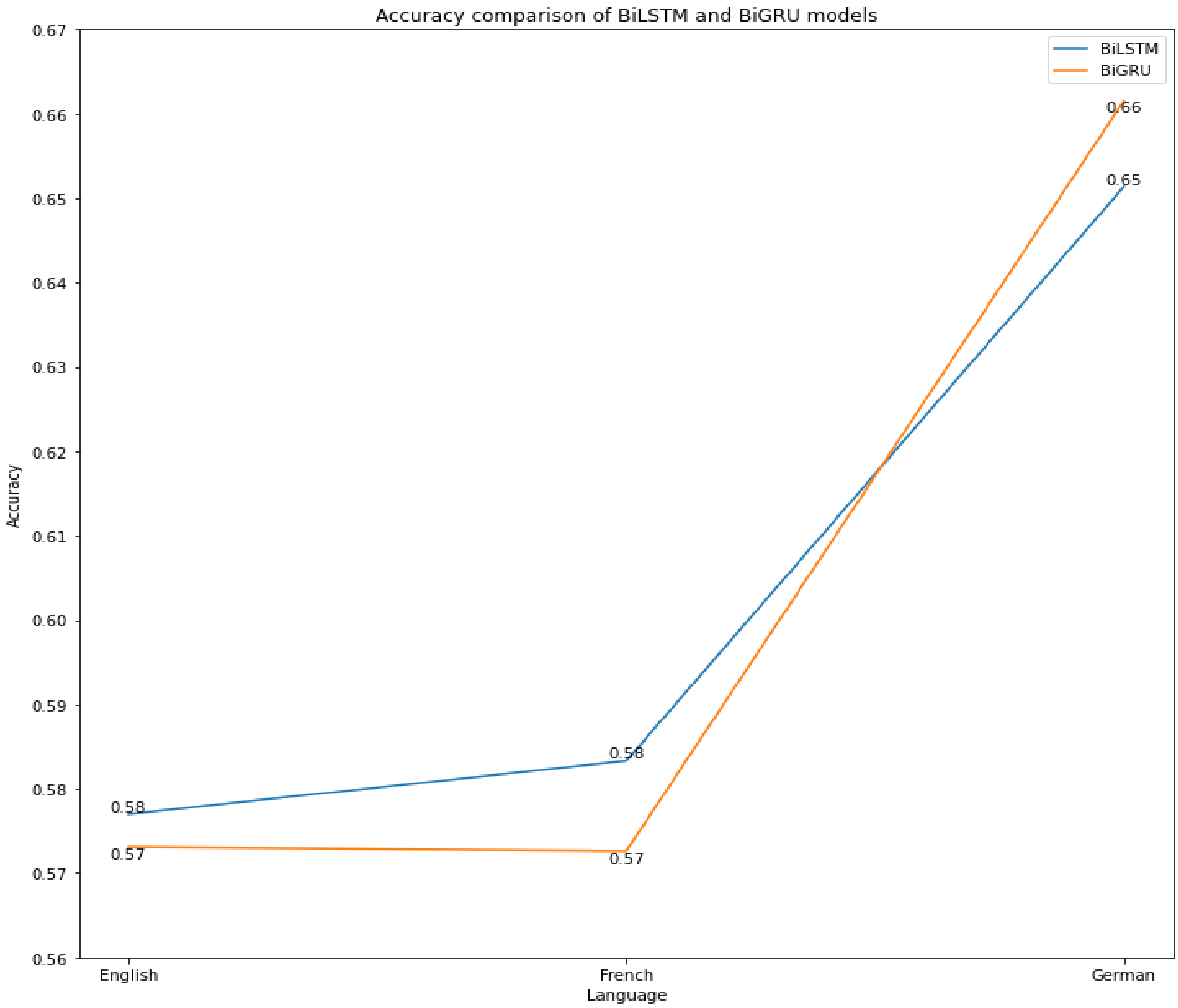


*Figure 5.1: Graph representing the performance of Bi-LSTM and Bi-GRU model across three languages*

## 6. Conclusion:

In conclusion, this research paper explored the effectiveness of the LESK approach and Bi-GRU/Bi-LSTM approach for word sense disambiguation. The results demonstrated that both methods contribute significantly to disambiguation accuracy. Further research can build upon these findings to enhance the performance of these approaches in real-world applications.

## 7. Future Works:

Current WSD methods are still not perfect, and there is always room for improvement. New methods could be developed that use more sophisticated features or that take into account more contextual information. Natural language data is often noisy and contains errors. This can make WSD challenging, as it can be difficult to determine the correct sense of a word when the context is ambiguous or when the data is corrupted. New methods could be developed that are more robust to noise and errors. Current WSD methods are typically developed for English. However, there is a need for WSD systems for other languages as well. New methods could be developed that are language-independent or that can be easily adapted to new languages. WSD is not only useful for machine translation and natural language understanding, but it can also be applied to other tasks, such as question answering, information extraction, and sentiment analysis. New applications for WSD could be explored.

## 8. References:

[1] Rentoumi, V., Giannakopoulos, G., Karkaletsis, V. and Vouros, G.A., 2009, September. Sentiment analysis of figurative language using a word sense disambiguation approach. In *Proceedings of the International Conference RANLP-2009* (pp. 370-375).

[2] Hung, C. and Chen, S.J., 2016. Word sense disambiguation based sentiment lexicons for sentiment classification. *Knowledge-Based Systems*, *110*, pp.224-232.

[3] Pamungkas, E.W. and Putri, D.G.P., 2017, February. Word sense disambiguation for lexicon-based sentiment analysis. In *Proceedings of the 9th International Conference on Machine Learning and Computing* (pp. 442-446).

[4] Jose, R. and Chooralil, V.S., 2015, November. Prediction of election result by enhanced sentiment analysis on Twitter data using Word Sense Disambiguation. In *2015 International Conference on Control Communication & Computing India (ICCC)* (pp. 638-641). IEEE.

[5] Seifollahi, S. and Shajari, M., 2019. Word sense disambiguation application in sentiment analysis of news headlines: an applied approach to FOREX market prediction. *Journal of Intelligent Information Systems*, *52*, pp.57-83.

[6] Calvo, H., Rocha-Ramirez, A.P., Moreno-Armendáriz, M.A. and Duchanoy, C.A., 2019. Toward universal word sense disambiguation using deep neural networks. IEEE Access, 7, pp.60264-60275.

[7] Nithyanandan, S. and Raseek, C., 2019. Deep learning models for word sense disambiguation: A comparative study. In Proceedings of the International Conference on Systems, Energy & Environment (ICSEE), Kerala, India.

[8] Saidi, R. and Jarray, F., 2022, March. Combining bert representation and pos tagger for arabic word sense disambiguation. In Intelligent Systems Design and Applications: 21st International Conference on Intelligent Systems Design and Applications (ISDA 2021) Held During December 13–15, 2021 (pp. 676-685). Cham: Springer International Publishing.

[9] Ahmed, M., Samee, M.R. and Mercer, R., 2018, December. A novel neural sequence model with multiple attentions for word sense disambiguation. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 687-694). IEEE.

[10] Wang, J., Zhang, Y., Yu, L.C. and Zhang, X., 2022. Contextual sentiment embeddings via bi-directional GRU language model. *Knowledge-Based Systems*, *235*, p.107663.

[11] Navigli, R., 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, *41*(2), pp.1-69.

[12] Stevenson, M. and Wilks, Y., 2003. Word sense disambiguation. *The Oxford handbook of computational linguistics*, *249*, p.249.

[13] Word-in-Context Dataset

[14] Shahi, T. B., Shrestha, A., Neupane, A., & Guo, W. (2020). Stock price forecasting with deep learning: A comparative study. *Mathematics*, *8*(9), 1441.

[15] Architecture of LSTM

[16] Architecture of GRU