International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Leveraging Contextual Factors for Word Sense Disambiguation in Hindi Language

Hirkani Padwad, Gunjan Keswani, Wani Bisen, Rajshree Sharma, Sopan Thakre, Aditi Tiwari

Submitted: 16/11/2023 **Revised:** 28/12/2023 **Accepted:** 08/01/2024

Abstract: This study presents an unsupervised model for addressing word sense disambiguation, to leverage accurate determination of the intended meaning of a word within a sentence. Identification of the correct sense demands high precision for applications like Machine translation, information retrieval, question answering, sentiment analysis, summarization, language generation. In recent years, few developments have been done in this field specifically for Indian languages. The unavailability of large labelled corpora poses a great challenge to applying large language models to this disambiguation task. Our approach leverages the deep learning BERT-based MuRIL model and measuring the Euclidean distance between synsets of words with multiple senses, achieving an accuracy of 89%. Second, we have curated a dataset based on the Indian theories of meanings which uses contextual factors for disambiguating the exact meaning of a word. The outcomes of this study offer valuable insights into the capabilities of language models applied to Indian languages, and their potential in reducing linguistic ambiguity.

Keywords: *Word Sense Disambiguation, MuRIL, INLTK.*

1. Introduction

In the domain of Natural Language Processing, mitigating ambiguity helps in multiple NLP tasks such as Machine Translation, Document Similarity measurement, Lexicography, Text mining, Question Answering systems, Document Summarisation and Cross Language Information Retrieval. The paper introduces a novel method for conducting Word Sense Disambiguation (WSD) specifically for Hindi language, which guides the procedure for determining the specific sense of a word used within a sentence, particularly in cases where the word exhibits multiple meanings due to polysemy. For example, word “आम ” (common/fruit name) can have various senses as per Indo Wordnet (a linked lexical knowledge base of wordnets of 18 scheduled languages of India).

Consider the following two sentences:

वह एक आम आदमी है |

बच्चे आम खा रहे हैं |

Here, we have used the word “आम” in both the sentences but their senses differ entirely. WSD in Hindi expects a sentence after which it suggests the sense of the ambiguous word on the basis of the context.

There are multiple approaches for WSD, irrespective of the language, all those can be categorized as Knowledge Based, Supervised, Semi supervised and Unsupervised. In this paper. we have used the hybrid approach which is based on the combination of Indo Wordnet Knowledge Base, MURIL model and our dataset. The popular BERT algorithm, also known as Bidirectional Encoder Representations from Transformers, is used by MURIL (Multilingual Representations of Indian Languages). Transformers is a deep learning model in which all output elements are connected to all input elements, and weightings between them are dynamically determined based on their connection. MuRIL is one of the latest approaches by Google that is trained for 17 Indian Languages. BERT helps in tokenising the Hindi sentence and hence helps in determining the correct sense of the word through embedding methodology.

Considering the Lesk algorithm, the foundation of this approach rested on the assumption that words within a specific vicinity would likely exhibit a shared thematic connection.

Unfortunately, Lesk's method is sensitive to the precise wording of dictionary definitions. Consequently, the presence or absence of a particular word can significantly influence the outcomes. Moreover, the algorithm only computes overlap among the explanations and instances of the senses under consideration.

Another algorithm, which is more accurate than Lesk is the Probabilistic Latent Semantic Analysis for Unsupervised Word Sense Disambiguation.

Probabilistic Latent Semantic Analysis (PLSA) serves as a statistical method employed to reveal concealed semantic patterns within a set of documents. Its primary application lies in topic modelling, wherein the objective is to unearth underlying topics that account for the trends in word distribution across the documents. Similarly, PLSA also

Department of Computer Science and Engineering Shri Ramdeobaba College of Engineering and Management, Nagpur
padwadhs@rknec.edu, keswanigv@rknec.edu, bisenwh@rknec.edu, sharmarh_1@rknec.edu, thakresp_1@rknec.edu, tiwarias_1@rknec.edu

gives rise to challenges such as susceptibility to overfitting (particularly when the number of topics exceeds the data size), issues with computational complexity, and the absence of dedicated handling for polysemy.

MURIL is designed to comprehend the context in which words appear, considering the surrounding words. PLSA treats documents as bags of words, ignoring the word order and context, which limits its understanding of the semantics. MuRIL helps in finding the most accurate meaning of the word by finding out the Euclidean distance between the synsets and the original word (from the sentence). This unique mechanism is based on handling multiple languages effectively due to its cross-lingual training. It generates compact word embeddings within a continuous vector space. These embeddings effectively encapsulate semantic connections among words, rendering them valuable for an array of subsequent tasks, including machine translation, sentiment analysis, and beyond.

The following contextual factors are listed by Bhartṛhari[16] for Sanskrit texts that help determine a word's exact meaning in the case of ambiguous sequences: association (saṃsarga), dissociation (viprayoga), companionship (sāhacarya), opposition (virodhitā), meaning [of co-occurring words]/purpose (artha), context (prakaraṇa), indication (liṅga), vicinity (saṃnidhi) of a specifying word, capacity (sāmarthya), propriety/suitability (aucitī), spatial context (deśa), temporal context (kāla), gender (vyakti), and accent (svara). Except for context, indication, spatial context, temporal context, gender, and accent—all of which are difficult to describe and some of which are irrelevant to Hindi—we chose a subset of these elements for our dataset compilation (COHin).

2. Related Work

Neural networks are extremely strong tools for tackling complicated issues because they are made to imitate the learning and pattern recognition capabilities of the human brain. Several algorithms have been proposed to aid in sense disambiguation, which is the process of determining the correct meaning of a word in a particular context. When a word has multiple meanings, the Lesk algorithm [1] aims to determine the most appropriate meaning based on the context in which the word appears. It uses dictionary definitions to compare the meanings of the word and selects the one that has the highest overlap with the context. The C fuzzy algorithm, also known as the fuzzy c-means algorithm [7], is a clustering technique used in data analysis and pattern recognition. It is based on fuzzy set theory and aims to partition a dataset into clusters by assigning membership degrees to data points rather than rigidly assigning them to a single cluster. The algorithm can handle noisy or uncertain data and provides a more flexible approach to clustering than traditional crisp clustering algorithms like k-means. In a similar way, PLSA [3] is a statistical model used for discovering latent semantic structures in text documents. It aims to represent words and documents in a lower-dimensional semantic space, capturing the underlying probabilistic relationships between them. PLSA uses the Expectation-Maximization algorithm to estimate model parameters and is particularly useful for tasks like document classification and information retrieval. Despite its limitations, PLSA has been widely used in various applications, including document classification, information retrieval, recommendation systems, and text mining. It provides a probabilistic framework for capturing the latent semantic structure of textual data, enabling more sophisticated analysis, and understanding of large document collections.

Association rule mining is a technique that has been used recently to find interesting relationships between items in a dataset. It finds item sets that are frequently used and uses those to derive association rules. These rules indicate the likelihood of certain items co-occurring together. Association rule mining has applications in market basket analysis, recommendation systems, and more. The structure of association rules is based on a left-hand antecedent and a right-hand consequent. The rule is stated as "If antecedent, then consequent." The rules indicate that if the antecedent items are present in a transaction or dataset, there is a high probability of the consequent items also being present. Different algorithms and variations, such as Apriori, FP-Growth, and Eclat, have been developed to efficiently mine association rules from large-scale datasets. These algorithms employ different strategies for itemset generation, pruning, and rule evaluation, aiming to optimise performance and scalability in different data scenarios.

The corpus-based Lesk algorithm [8] is an extension of the original Lesk algorithm that incorporates a corpus, a large collection of text, to improve word sense disambiguation. It was developed to address some of the limitations of the traditional Lesk algorithm, which relies solely on dictionary definitions. It leverages the context in which an ambiguous word appears by analysing its surrounding words in a corpus. By comparing the overlap of word senses between the ambiguous word and the words in its context, the algorithm calculates a score and aims to select the sense with the greatest score.

The Word co-occurrence [8] approach refers to the frequency or occurrence of words appearing together within a specific context or proximity in a text corpus. It is a fundamental concept in natural language processing and computational linguistics that captures the relationship and patterns between words based on their contextual usage. In the context of word co-occurrence, a corpus is typically analysed to determine the frequency of pairs or groups of words appearing together. The co-occurrence can be measured at different levels, such as within a sentence, a paragraph, or a larger window of words.

Furthermore, Google AI created the ground-breaking language model BERT (Bidirectional Encoder Representations from Transformers) [9][10]. It introduced the concept of pretraining and fine-tuning to achieve state-of-the-art results in various natural language processing tasks. BERT is based on a transformer architecture, a deep learning model that allows for efficient processing of sequential data. The primary innovation of BERT is its bidirectional training approach, which allows it to understand a word's context by considering both words that come before and after it. To learn word representations, BERT is pre-trained on copious amounts of text data from sources like books and websites. Similarly, IndicBERT is a language model specifically developed for Indic languages, which are spoken in the Indian subcontinent. It is derived from the BERT (Bidirectional Encoder Representations from Transformers) model and tailored to handle the complexities and nuances of Indic languages. IndicBERT shares the same architecture as BERT but undergoes additional training on large amounts of text data from Indic languages. Its ability to handle the complexities of Indic languages and its transfer learning capabilities make it a valuable tool for researchers, developers, and organisations working with Indic language data and applications.

Among the most recent models for WSD is MuRIL [11], is a state-of-the-art language model developed by Google Research India. It is designed to understand and generate text in multiple Indian languages, including Hindi, Bengali, Tamil, Telugu, and more. The main objective of MuRIL is to bridge the gap in natural language processing research for Indian languages, which often face challenges due to limited linguistic resources and diverse language structures. Similar to models like BERT, MuRIL is based on a transformer-based architecture, but it is specially designed to manage the subtleties and complexities of Indian languages. MuRIL has been trained on a vast quantity of multilingual textual data from the internet, spanning various genres and domains. This training process enables the model to learn contextual representations of words and sentences, capturing their semantic meaning and syntactic relationships within the context. One of the key strengths of MuRIL is its ability to handle code-switching, a common phenomenon in Indian languages where multiple languages are mixed within a single sentence or conversation. MuRIL can effectively understand and process code- switched text, enabling more accurate and meaningful language understanding and generation. Moreover, MuRIL incorporates pre-training objectives that focus on specific linguistic tasks, such as masked language modelling and sentence order prediction. These objectives enhance the model's ability to understand the structures and semantics of Indian languages. In conclusion, neural networks and related algorithms have significantly advanced our ability to understand and process complex data, ranging from text analysis to image recognition. They have become essential tools in various fields, driving innovation and progress in artificial intelligence research and applications.

Sentiment analysis and word sense disambiguation for Hindi language machine translation was combined in another method [13] of Rule Based Fuzzy Computing Approach on Self-Supervised Sentiment Polarity Classification with Word Sense Disambiguation in Machine Translation. It gained another depth with the concepts of fuzzy rules and the use of three different kinds of lexicons. An unsupervised graph-based approach for Hindi word sense disambiguation applying a random walk on the graph has been used in [14] achieving an average accuracy of 72.09%.

3. Proposed Method

We present a hybrid approach in which we obtain best possible sense from among all possible senses available in Indo word net and then later we would pass these results to pretrained MuRIL model. We further fine tune the MuRIL model using the curated dataset.

A. Dataset Generation

We compiled the CoHin dataset for Hindi on the basis of the contextual factors identified by Bhartṛhari. The dataset consists of following information.a. Text: It contains the actual text segment in Hindi where an ambiguous word occurs. The ambiguous word is highlighted with '*' before the word.

b. Type of contextual factor: Second column indicates which contextual factor applies to each text segment. List of contextual factors under consideration is as follows.

1. Association (saṃsarga): This indicates whether the contextual factor of "Association" applies to the ambiguous sequence in the given text segment.

2. Dissociation (viprayoga): Like the Association, this indicates whether the contextual factor of "Dissociation" applies.

3. Companionship (sāhacarya): Indicates if "Companionship" is a relevant contextual factor.

4. Opposition (virodhitā): Indicates if "Opposition" is a relevant contextual factor.

5. Meaning/Purpose (artha): Indicates if "Meaning/Purpose" is a relevant contextual factor.

6. Capacity (sāmarthya): Indicates if "Capacity" is a relevant contextual factor.

7. Propriety/Suitability(aucitī): Indicates if "Propriety/ Suitability" is a relevant contextual factor.

c. Related segment: Contains the parts of the text which exhibit the applicable relationship.

Table 1 shows a sample of the CoHin dataset.

Text	Type of contextual factor	Related Segment
bachade ke sAth *__dhenu__* jA rahi hai	1	Bachade ke sAth
bachade ke binA *__dhenu__* jA rahi hai	2	Bachade ke binA
__rAm__ lakshaman ne yudHa jeetA	3	lakshaman
Karan aur *__arjun__* unke sAthiyo ki pratikshA kar rahe hai	4	Karan
Hum *__shiva__* ki ArAdhanA karte hai	5	ArAdhanA karte hain
__madhu__ kokilA ko madahosh kar detA hai	6	Madahosh kar detA hai
adhikAriyo ne ache khilAdiyo kA *__paksha__* lenA chAhiye	7	khilAdiyo kA

Table 1: Sample of CoHin dataset

Hindi texts were obtained from multiple sources including Hindi Corpora by CFILT[1], Wikipedia. The texts were labelled and verified by manual annotation. Till now, the total number of samples in the dataset are 2,000. Number of samples belonging to each category are around 200 to 300, thus maintaining a balance in the dataset. Currently, we have tagged a single text sample with only one type of relationship at a time. In future, with availability of more resources, we plan to extend the dataset with more samples to make effective use of Large Language Models.

We also plan to apply unsupervised methods to a huge unlabelled hindi corpora containing similar kind of patterns to our CoHin samples to experiment with low resource setting.

B. Data Extraction

To extract possible senses of an ambiguous word, we needed a source which would contain word-level semantic information. So, we have used the Centre for Indian Language Technology's (CFILT's) IndoWordNet[2] for Hindi to extract information about ambiguous words. The 18 scheduled languages of India—Assamese, Bangla, Bodo, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Malayalam, Meitei, Marathi, Nepali, Odia, Punjabi, Sanskrit, Tamil, Telugu, and Urdu—are represented in this enormous linked lexical knowledge base (LKB) of wordnets. This knowledge base inputs a word and returns synsets for that word. A typical synset object consists of the following attributes:

1. Synset ID
2. POS (part of speech)
3. Synonyms (a list)
4. Gloss (meaning in Hindi)
5. Example statement
6. Gloss in English

To access IndoWordNet, we have used a python-based API, pyiwn [15], provided by IIT Bombay.

C. WSD Algorithm

The algorithm proposed here is broken down into following steps:

Step 1: Input preprocessing

A sentence containing an ambiguous word is input to the model. The input sentence is preprocessed by applying tokenization and the words are reduced to their root form using indic-NLP library.

Following is an example of an input sentence and the result of preprocessing.

Input: मुझे खाना बहुत पसंद है।

Sentence tokens after preprocessing: ['मुझे', 'खाना', 'बहुत', 'पसंद', 'है']

From these tokens any one word is selected to identify its correct sense.

Step 2: Synonym set generation

Synonym set is a database of all possible senses for the ambiguous word. With root word argument, IndoWordNet returns a group of synsets pertaining to the input. From each of these synsets, *synset.head_word()* yields the head word out of a list of words for this synset. For each of these synsets, we calculate their first example sentence's cosine similarity with the input sentence. Cosine similarity is the measure of relatedness of two vectors and lies in the range [0, 1]. Out of all these synonyms, we filter out the 7 best senses (yielding maximum similarity), if there are at least seven of them. This is our synonym set.

So, we generated synonym set for word: {'ख़ाना', 'भक्षण', 'भोजन', 'पीना', 'चाटना', 'दराज़', 'अहारना', 'जीमना', 'खाद्य वस्तु', 'बर्बाद करना', 'आहार'}

[1] https://www.cfilt.iitb.ac.in/download_new.html

[2] https://www.cfilt.iitb.ac.in/wordnet/webhwn/

Step 3: Highly similar sentences generation

Now we generate a list of sentences, with ambiguous words replaced by each of the words from the synonym set. This is a sense-replaced sentences list.

Again, sentences are filtered out. For each of these sentences from the sense-replaced sentences list, their cosine similarity with input sentence is calculated and only the sentences with at least 0.9 similarity are moved forward. This is a highly-similar sentences list.

Step 4: Preprocessing

In this step, stop words are removed. The sentences from the highly similar sentences list are tokenized and stopwords, if any, are removed from each sentence. We have used *stopwords.txt*[link] to identify stopwords in Hindi. Resultantly, we obtain a pre-processed highly-similar sentences list.

The result that we get for above input example is as follows.

[

'मुझे **आहार** बहुत पसंद है',

'मुझे **बर्बाद** **करना** बहुत पसंद है',

'मुझे **भोजन** बहुत पसंद है'

]

Step 5.1: Using Pretrained Model MuRIL

MuRIL is a BERT model pretrained on 17 Indian languages. The reason of choosing this is that this is solely meant for IN languages and so includes larger corpus of data. MuRIL which stands for Multilingual Representations for Indian Languages is pretrained on monolingual as well as parallel segments thus uses both Translation and Transliteration of Pair of words. It compares two words on these grounds and so gives relation between the two.

In our approach we use the transliteration power of MuRIL. The list of highly similar sentences that we get as a result of INLTK and preprocessing is input to MuRIL, that checks whether the word in place of ambiguous word fits in the sentence or not, by comparing the replaced word with its surrounding words.

So, for our example:

आहार - पसंद	: 0.011560117825865746
बर्बाद - पसंद	: 0.008085211738944054
भोजन - पसंद	: 0.010638106614351273

Step 5.2: Fine tuning MuRIL model

Further, we fine-tuned the MuRIL model using our dataset COHin in order to rule out unobvious senses of a word. After fine tuning the model, the following results were obtained.

आहार - पसंद : 0.125746238862142562

बर्बाद - पसंद : 0.002526688741123659

भोजन - पसंद : 0.099528462358745923

This approach generates word embeddings and determines the distance between them. The one with least distance is selected as answer.

In the stated example, आहार is selected as final sense for the ambiguous word खाना.

tep 6: Best sense selection and output

The word that we selected as best fit for given input is replaced with ambiguous word in input sentences.

As a result,

Input : मुझे खाना बहुत पसंद है।

Output: मुझे आहार बहुत पसंद है।

Input sentences	Correctly disambiguated	Accuracy
200	176	88%

Table 2: Accuracy

Input Hindi sentence	Google translation	Correct translation
इस तीर का फल बहुत नुकीला है	The fruit of this arrow is very sharp	The tip of this arrow is very sharp
कर भला सो हो भला	Do well sleep well	As you sow, so you reap
वह पेड़ के निचे पत्ते खेल रहे हैं	They are playing leaves under the tree	They are playing cards under the tree

Table 3: Google translation vs correct translation

Methods studied for Hindi WSD	Accuracy (%)	Precision	Recall	F-measure
Lesk Algorithm	40–70	-	-	-

Probabilistic Latent Semantic Analysis	74.12	-	-	-
Association Rules	72	-	-	-
Corpus-based lesk algo	65.17	-	-	-
Word co-occurrence	-	68.73	64.41	-
IndicBERT		-	-	60.59
mBERT				61.29

Table 4: HWSD methods comparison

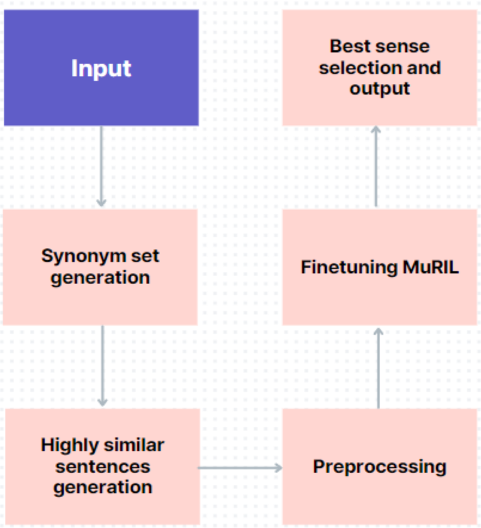

Input
Best sense selection and output
Synonym set generation
Finetuning MuRIL
Highly similar sentences generation
Preprocessing

Fig 1: Model workflow

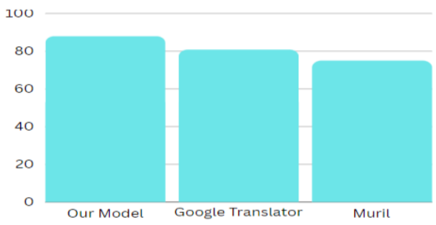


Fig 2: Accuracy comparison

4. Results and Analysis

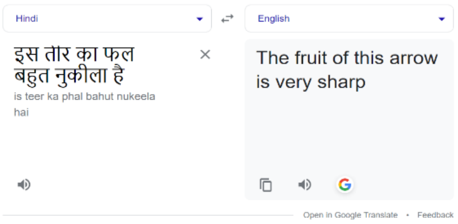
MuRIL employs a composite of loss functions, including language modeling, masked language modeling, and next sentence prediction. Unlike smaller neural networks, it's not strictly bound by training epochs, as it continuously learns from a continuous stream of data. It often utilizes large batch sizes, ranging from hundreds to thousands, with smaller batches used in fine-tuning. MuRIL adopts a low initial learning rate (typically between 1e-4 and 1e-5), possibly with a scheduled change. The model's size depends on architecture and available resources and is trained on extensive multilingual text from the internet.

We tested our model against 200 inputs. Table 2 summarizes the result of our algorithm.

Also, there are instances when Google incorrectly translated ambiguous Hindi sentences as shown in Table 3. However, for some of these sentences (Figure 4), when fed with the result generated from our model, it translated it correctly implying that the model acted as an intermediary between argument sentences and Google translate and increased the accuracy of Google translator.

The model relies on IndoWordNet to generate synsets for ambiguous words. So, if some word is not found in the knowledge base, the model returns an error. Also, for ambiguous words whose synonym set consists of just the root word, the input sentence is returned as it is.

इस तीर का फल बहुत नुकीला है


Hindi
English

इस तीर का फल बहुत नुकीला है
is teer ka phal bahut nukeela hai

The fruit of this arrow is very sharp

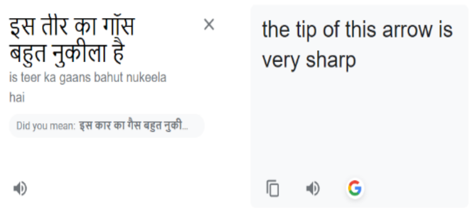
Open in Google Translate • Feedback

Fig 3: Incorrect Google translation

Google translated this incorrectly so we pushed the Hindi sentence into our WSD Algorithm from which we get another refined Hindi sentence.

इस तीर का गाँस बहुत नुकीला है

And now the Google translator can give accurate translation of this.


इस तीर का गाँस बहुत नुकीला है
is teer ka gaans bahut nukeela hai

Did you mean: इस कार का गैस बहुत नुकी...

the tip of this arrow is very sharp

Fig 4: Correct translation by Google using our model's output

To assess the effectiveness of the model created for our objective of disambiguation, we have chosen the similarity or relatedness metric in this instance. We used a word similarity dataset created by IIIT Hyderabad for Indian languages [12] which is evaluated by human annotators. It includes a database of similarity results for 200-word pairs across 7 Indian languages. In this job, we assess how effectively our model captures similarity as measured by the similarity datasets. The two measurements are compared using the Pearson correlation after the cosine similarity of the words is computed. The quality of the embeddings increases with correlation values that approach 1.

Model	Word Similarity (Pearson correlation)
INLP	0.626
FT W	0.575
FT WC	0.551
Our model	0.60

Table 5: Pearson correlation for various models

5. Conclusion and Future Scope

The increased computational resources could expedite processing, yielding faster results. Integrating diverse data sources beyond CFILT could further enhance the model's performance.

In conclusion, this paper presents a novel approach to Word Sense Disambiguation based on a hybrid approach using MuRIL (BERT)-based fine-tuned model. Our approach showcases significant accuracy in disambiguating multi-sense words, contributing to the fields of language processing and translation. The potential for future enhancements, incorporating additional data sources, opens avenues for even more effective WSD solutions. Looking ahead, there are promising directions for enhancing this work. Extending the model to handle code mixed-language sentences, specifically integrating English and Hindi, would align with real-world language use and further elevate its practicality. Leveraging existing algorithms designed by Google for Indian Regional Languages could provide a strong foundation for such advancements. Representation of common-sense knowledge and world knowledge can further improve the disambiguation task by ruling out some obvious looking senses which in fact may be wrong. With availability of more resources, we plan to extend our dataset to incorporate world knowledge across multiple domains to leverage the power of state-of-the-art language models in an optimum way. Analysis of the wrong predictions of the current model and use of the newest feature extraction techniques in a sequential way may further improve the results.

References

[1] Lesk M., “Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone” in Proceedings of the 5th Annual International Conference on Systems Documentation, Ontario, Canada, pp. 24-26, 1986.

[2] Baldwin T., Kim S., Bond F., Fujita S., Martinez D., and Tanaka T., “A Reexamination of MRDbased Word Sense Disambiguation,” Journal of ACM Transactions on Asian Language Processing, vol. 9, no. 1, pp. 1-21, 2010.

[3] Gaurav S Tomar, Manmeet Singh, Shishir Rai, Atul Kumar, Ratna Sanyal and Sudip S, “Probabilistic Latent Semantic Analysis for Unsupervised Word Sense Disambiguation” in International Journal of Computer Science Issues, Vol. 10, Issue 5, 2013

[4] Banerjee S. and Pederson T., “An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet,” in Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing, Mexico City, Mexico, pp. 136-145, 2002.

[5] Banerjee S. and Pederson T., “Extended Gloss Overlaps as a Measure of Semantic Relatedness,” available at: http://www.d.umn.edu/~tpederse/ Pubs/ijcai03.pdf, last visited 2013.

[6] Vasilescu F., Langlasi P., and Lapalme G., “Evaluating Variants of the Lesk Approach for Disambiguating Words,” available at: http://www. lrec-conf.org/proceedings/lrec2004/pdf/219.pdf, last visited 2012.

[7] Zhang, D. Q., Chen, S. C. (2003), “Clustering incomplete data using kernel-based fuzzy c-means algorithm”, Neural Processing Letters, 18 (3) 155-162.

[8] Satyendr Singh and Tanveer Siddiqui, “Utilizing Corpus Statistics for Hindi Word Sense Disambiguation”, In The International Arab Journal of Information Technology, Vol. 12, No. 6A, 2015

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. “Bert: Pre-training of deep bidirectional transformers for language understanding”, arXiv preprint arXiv:1810.04805

[10] Luyao Huang, Chi Sun, Xipeng Qiu∗ , Xuanjing Huang, “GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge”, In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3509–3514, Hong Kong, China, November 3–7, 2019

[11] Simran Khanuj, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar

Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, Partha Talukdar, “MuRIL: Multilingual Representations for Indian Languages”, In arXiv:2103.10730v2 [cs.CL] 2 Apr 2021

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. “Bert: Pre-training of deep bidirectional transformers for language understanding”, arXiv preprint arXiv:1810.04805

[13] Luyao Huang, Chi Sun, Xipeng Qiu∗ , Xuanjing Huang, “GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge”, In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3509–3514, Hong Kong, China, November 3–7, 2019

[14] Simran Khanuj, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, Partha Talukdar, “MuRIL: Multilingual Representations for Indian Languages”, In arXiv:2103.10730v2 [cs.CL] 2 Apr 2021

[15] Akhtar, S.S., Gupta, A., Vajpayee, A., Srivastava, A., Shrivastava, M., 2017,pp. In: ‘‘Word similarity datasets for Indian languages: Annotation and baseline systems. Association for Computational Linguistics, Valencia, Spain, pp. 91–94.

[16] Mishra, B. K., & Jain, S. (2023). An Innovative Method for Hindi Word Sense Disambiguation. SN Computer Science, 4(6), 704.

[17] P. Jha, S. Agarwal, A. Abbas and T. Siddiqui, "Comparative Analysis of Path-based Similarity Measures for Word Sense Disambiguation," 2023 3rd International conference on Artificial Intelligence and Signal Processing (AISP), VIJAYAWADA, India, 2023, pp. 1-5, doi: 10.1109/AISP57993.2023.10134960.

[18] Ritesh Panjwani, Diptesh Kanojia, and Pushpak Bhattacharyya, pyiwn: A Python-based API to access Indian Language WordNets, Global WordNet Conference (GWC 2018), January 2018.

[19] Emilie Aussant. Sanskrit Theories on Homonymy and Polysemy . Bulletin d’Études Indiennes, 2014, Les études sur les langues indiennes. Leur contribution à l’histoire des idées linguistiques et à la linguistique contemporaine, 32. ffhalshs-01502381f