PARIBHAASHA

SHUBHA JAIN
Associate Professor
Department of Computer Science Engineering Axis Institute of Technology and Management, Kanpur, U.P.

Nikhil Mishra || Rishit Maurya || Aman Prajapati || Harshit Kumar Nishad
Student
Department of Computer Science Engineering (AIML) Axis Institute of Technology and Management, Kanpur, U.P.

ABSTRACT

As in world increasingly reliant on digital communication, effective writing skills are paramount. However, for "Hindi" language users, finding tools that take their unique linguistic needs presents a challenge. This abstract will come with a novel solution: a Grammarly-like platform precisely designed to empower Hindi writers by offering advanced language correction and enhancement features tailored specifically for the nuances of the Hindi language. At the core of our platform lies a sophisticated text analysis engine, powered by state-of-the-art natural language processing algorithms meticulously trained on vast corpora of Hindi text.
Our platform offers a seamless user experience, with a user-friendly interface designed to facilitate effortless interaction in single time of corrections. Users can simply input their "Hindi" text into the platform, and within moments, receive comprehensive feedback and suggestions aimed at refining their writing to perfection. In addition to its high dynamic language correction capabilities, our platform also features advanced plagiarism detection functionality. Leveraging cutting-edge algorithms that is particular focusing on platform thoroughly scans user-submitted text against an extensive database of online sources, academic papers, and published works to ensure originality and integrity, vulnerable and fully secured. Furthermore, our platform goes beyond mere error correction to offer personalized writing insights and guidance. Users can set custom writing goals based on their intended audience, style preferences, and communication objectives. Our platform then provides tailored suggestions and recommendations aimed at helping users achieve their desired writing outcomes, whether it be crafting formal business documents, creative literature, or casual social media posts.
This engine enables our platform to accurately identify and rectify a myriad of grammatical errors, spelling mistakes, and stylistic inconsistencies commonly encountered in Hindi writing.

INDRODUCTION

In recent years, advancements in the field of natural language processing (NLP) , artificial intelligence (AI) and LLM models have significantly influenced various domains, including linguistics, education, and digital communication. Grammarly, an AI-driven writing assistant, has become a prominent tool in enhancing English language writing by providing real-time grammar, punctuation, and style corrections But not any portal will work in Hindi correction. Its widespread adoption underscores the increasing demand for automated language assistance in a globalized world. As despite efficacy in English correction (grammerly used), there is a notable absence of similar tools tailored for other languages, particularly Hindi, which is the fourth most spoken language globally. The integration of “Paribhaasha” technology for the Hindi language presents unique challenges and opportunities. Hindi, with its rich linguistic heritage and diverse dialects, poses complex syntactic and semantic structures that require sophisticated NLP models with the integrated AI models. Furthermore, the use of “Devanagari” script (that is pre developed script) adds another layer of complexity in terms of character recognition and processing. Addressing all these challenges can pave the way for developing a robust Hindi language writing assistant, that will help in user writing experience and which can benefit millions of native speakers, learners, and professionals. This research aims to explore the feasibility of developing a “Paribhaasha” tool for Hindi. It will examine the current state of Hindi NLP technologies and works on current user written Hindi sentence and fix all the corrections on the basis of grammatical and sentence correction. As identify the gaps and limitations, and propose methodologies to overcome these hurdles. By leveraging recent advancements in machine learning and AI, this study process to contribute to the growing field of automated language processing and enhance digital literacy in the Hindi-speaking community.

METHODOLOGY

The development of a “Paribhaasha” tool for the Hindi language involves a multi-faceted approach combining various natural language processing (NLP) techniques, machine learning models, and linguistic resources. This section outlines the comprehensive methodology employed in the creation and evaluation of the proposed tool.

1. Data Collection and Pre-processing:

A large and diverse corpus of Hindi text is essential for training effective NLP models. We compile this corpus from multiple sources, including newspapers, books, online articles, social media, and user-generated content. The corpus should encompass various dialects, styles, and contexts to ensure robustness and versatility.

2. Linguistic Resource Development:

Develop a comprehensive lexicon of Hindi words, including their morphological variants, synonyms, and antonyms. Establish a set of grammar rules that capture the syntactic and semantic nuances of Hindi. These rules are crucial for identifying grammatical errors and suggesting corrections. Implement dependency parsing to understand the grammatical relationships between words in a sentence. This helps in detecting complex grammatical errors and understanding sentence structure.

3. Model Training:

Train machine learning models using the annotated corpus. Models such as Conditional Random Fields (CRFs) and Recurrent Neural Networks (RNNs) are employed for tasks like part-of-speech tagging and error detection. Utilize transfer learning techniques by fine-tuning pre-trained models (e.g., BERT) on the Hindi corpus to enhance performance on specific tasks. Develop classifiers for different types of grammatical errors, such as subject-verb agreement, tense consistency, and punctuation. Implement sequence-to-sequence models for generating corrections. These models take erroneous sentences as input and output corrected sentences.

4. Evaluation and Optimization

Use standard NLP evaluation metrics, such as precision, recall, and F1-score, to assess the performance of error detection models. Evaluate correction models based on BLEU (Bilingual Evaluation Understudy) scores and human judgment to ensure the suggested corrections are contextually appropriate and grammatically accurate. Deploy a beta version of the tool to a group of Hindi speakers for real-world testing. Collect user feedback on the accuracy and usability of the tool. Use this feedback to iteratively improve the models and fine-tune the error correction algorithms.

5. Integration and Deployment

Design an intuitive user interface that allows users to input text, receive real-time feedback, and view suggested corrections. Ensure the interface supports the "Devanagari" script and offers functionalities like highlighting errors and providing explanations. Deploy the tool as a web-based application and a browser extension to reach a wide audience. Set up a system for continuous monitoring and updating of the models based on new data and user interactions

LITERATURE SURVEY

This literature survey reviews the relevant studies and technologies that underpin the creation of such a tool, focusing on Hindi language processing, grammar checking systems, and related AI advancements. The development of a "Paribhaasha" tool for the Hindi language requires a thorough understanding of existing research in natural language processing (NLP), machine learning, and linguistic analysis

1. Natural Language Processing for language correction:

Several studies have addressed Hindi's morphological complexity. Shrivastava and Bhattacharyya (2008) developed a morphological analyzer for Hindi that handles inflectional and derivational morphology. More recent work by Kumar et al. (2016) focuses on creating robust morphological analyzers using machine learning techniques, which are essential for understanding and processing Hindi text.

2. Grammar Checking Systems

General Grammar Checkers, checking systems have been extensively researched for English, with tools like "Paribhaasha", Microsoft Word's grammar checker, and the Language Tool project offering comprehensive solutions. These systems typically employ rule-based, statistical, and hybrid approaches to detect and correct grammatical errors. Efforts to develop grammar checkers for Hindi have been relatively limited. The work by Singh et al. (2010) represents one of the early attempts to create a rule-based grammar checker for Hindi, focusing on common grammatical errors such as subject-verb agreement and punctuation. More recent research by Goyal and Lehal (2020) explores machine learning approaches to enhance the accuracy and coverage of Hindi grammar checkers.

3. Machine Learning and Deep Learning Approaches

Traditional machine learning models, such as Conditional Random Fields (CRFs) and Support Vector Machines (SVMs), have been successfully applied to NLP tasks, including part-of-speech tagging and named entity recognition for Hindi.
The advent of deep learning has revolutionized NLP. Models such as Long Short-Term Memory (LSTM) networks and Transformers have shown remarkable success in various languages. Pre-trained models like BERT (Bidirectional Encoder Representations from Transformers) and its variants have been fine-tuned for specific tasks, including error detection and correction. BERT's application to Hindi has been demonstrated in studies that adapt the model to understand and generate Hindi text effectively.

4. Evaluation and User Interaction

Standard evaluation metrics for grammar checking systems include precision, recall, and F1-score. Human evaluation is also critical, as automated metrics may not fully capture the context and appropriateness of corrections. User interaction and feedback are crucial for the success of language tools. Research by Farzindar and Inkpen (2015) emphasizes the importance of intuitive interfaces and real-time feedback in enhancing user engagement and learning. Studies on user experience with grammar checkers highlight the need for tools that are not only accurate but also user-friendly and educational

5. Ethical Considerations

Data privacy is a significant concern in NLP applications. Ensuring user data protection and compliance with privacy regulations is essential for gaining user trust and adoption NLP models are susceptible to biases present in training data. Research by Bolukbasi et al. (2016) and others underscores the importance of developing fair and unbiased models, which is particularly relevant for creating inclusive and equitable language tools .

PROPOSED SYSTEM

The proposed system aims to develop a Paribhaasha tool for the Hindi language, designed to detect and correct grammatical errors, provide style suggestions, and enhance overall writing quality. The system leverages natural language processing (NLP) techniques, machine learning models, and linguistic resources tailored specifically for Hindi. This section details the architecture, components, and functionalities of the proposed system.

1. System Architecture

Input Module: Accepts Hindi text input from the user.
Preprocessing Module: Cleans and normalizes the input text.
Error Detection Module: Identifies grammatical errors using machine learning models and linguistic
Error Correction Module: Suggests corrections for detected errors.
Style and Clarity Module: Provides suggestions for improving writing style and clarity.
User Interface Module: Displays errors and suggestions to the user in an intuitive manner.

2. Data Collection and Preprocessing

A large and diverse corpus of Hindi text is compiled from sources such as books, news articles, social media, and user-generated content. The corpus covers various dialects, genres, and contexts to ensure comprehensive language coverage.The text data is cleaned to remove noise, such as irrelevant symbols and formatting issues. This step ensures that the input to the NLP models is consistent and reliable.
Text normalization includes tokenization, stemming, and lemmatization. This process converts the text into a standardized form, making it easier to analyze and process.

3. Error Detection

A set of predefined grammatical rules specific to Hindi is used to detect common errors such as subject-verb agreement, tense consistency, and punctuation errors. These rules are derived from linguistic studies and expert knowledge of Hindi grammar. Supervised learning models, including Conditional Random Fields (CRFs) and Long Short-Term Memory (LSTM) networks, are trained on the annotated corpus to identify grammatical errors. Transfer learning with pre-trained models like BERT is employed to enhance error detection accuracy. A hybrid approach combines rule-based and machine learning methods to leverage the strengths of both. Rules provide precision in detecting well-defined errors, while machine learning models offer flexibility and adaptability in identifying complex and context-dependent errors.

4. Error Correction

Sequence-to-sequence models, particularly Transformer-based models, are used to generate corrections for the identified errors. These models take the erroneous sentence as input and produce a grammatically correct version as output. Corrections are generated considering the context of the sentence to ensure they are appropriate and maintain the intended meaning. This involves using attention mechanisms in Transformer models to capture context effectively.

5. User Interface

The user interface highlights detected errors in the input text and displays suggestions for corrections. Errors are color-coded based on their type (e.g., grammar, style, clarity) to help users quickly identify and understand issues. Users can interact with the suggestions, accepting or rejecting them. This feedback loop helps the system learn from user preferences and improve its future suggestions.

6. Evaluation and Continuous Improvement

The system's performance is evaluated using standard NLP metrics, including precision, recall, and F1-score, to measure the accuracy of error detection and correction. Continuous feedback from users is collected to refine the models and rules. User satisfaction and improvement in writing quality are key indicators of the system's effectiveness. The system is regularly updated with new data and improved algorithms to keep pace with evolving language usage and maintain high performance.

IMPLEMENTATION AND RESULTS

Implementation

The implementation of Paribhaasha tool for the Hindi language involves several stages, including data preparation, model training, error detection and correction, user interface development, and system integration. This section outlines the detailed steps and technologies used in each phase of the implementation process.

1. Data Preparation

- Corpus Compilation

Text data is collected from diverse sources such as newspapers (e.g., Dainik Bhaskar), books, online articles, social media platforms (e.g., Twitter, Facebook), and user-generated content. A corpus of approximately 10 million sentences is compiled to cover various dialects, styles, and contexts.

- Data Cleaning

Noise Removal: Scripts are written in Python to remove HTML tags, special characters, and other noise. Normalization: The indic-nlp-library is used for tokenization, stemming, and lemmatization of Hindi text.

2. Model Training

- Part-of-Speech Tagging

Model: Conditional Random Fields (CRFs) are used for part-of-speech tagging.
Training: The CRF model is trained using the manually annotated corpus. Libraries like are utilized for this purpose. Dependency parsers based on the Universal Dependencies framework are implemented. Training: Models are trained using annotated data to understand syntactic structures and relationships between words.

Rule-based Detection: Grammar rules are encoded using Python scripts, focusing on common errors such as subject-verb agreement and incorrect tense usage. BERT: Fine-tuning of multilingual BERT (mBERT) on the annotated Hindi corpus for error detection tasks. Implementation: Using the transformers library by Hugging Face, the mBERT model is fine-tuned on error detection tasks. Sequence-to-Sequence Models: Transformer models are used for generating corrections. Training: The models are trained on pairs of erroneous and corrected sentences using the fairseq library

- **Error Detection and Correction**

Hybrid Approach
Rule-based methods and machine learning models are integrated to ensure comprehensive error detection and correction. The rule-based system first filters easily detectable errors, followed by the application of the machine learning model for more complex errors.

- **Context-Aware Correction**

Transformer models employ attention mechanisms to consider the context of errors.
Implementation: Using the transformers library, attention layers are fine-tuned to prioritize contextual understanding during correction generation.

- **User Interface Development**

Front-End Design
Framework: The user interface is built using React.js for a responsive and interactive user experience. Error highlighting, correction suggestions, and explanations are implemented. Errors are color-coded based on their type (e.g., grammar, style, clarity).

- **Back-End Integration**

Server: The back-end is developed using Flask, providing a REST API for the front-end to interact with. Deployment: The system is deployed on cloud platforms like AWS, ensuring scalability and reliability.

- **Real-Time Feedback**

Implementation: WebSocket connections are used to provide real-time feedback and suggestions as users type.

- **Automated Evaluation**

Metrics: Precision, recall, and F1-score are calculated using a separate test set of annotated sentences.
Tools: scikit-learn is used for evaluating classification models and generating reports.

- **Human Evaluation**

Human evaluators assess the accuracy and usefulness of the suggestions.
Evaluators provide feedback on the appropriateness of corrections, clarity of explanations, and overall user satisfaction.

- **Continuous Improvement**

User Feedback Loop
Feedback is collected through the user interface where users can rate suggestions and provide comments. Feedback is analyzed using NLP techniques to identify common issues and areas for improvement.

- **Model Updates**

Models are periodically retrained with new data and user feedback to improve performance.
Updated models are deployed using continuous integration/continuous deployment (CI/CD) pipelines, ensuring seamless updates.

- **Ethical Considerations**

Compliance: Ensure compliance with data protection regulations such as GDPR.
Anonymization: User data is anonymized to protect privacy before being used for model training. Evaluation: Regularly evaluate models for bias and ensure fairness across different user demographics.
Correction: Implement techniques to mitigate identified biases, ensuring the tool provides equitable assistance to all users.

Results

The development and evaluation of the proposed "Paribhaasha" tool for Hindi yielded promising results. This section presents the findings from both automated and human evaluations, demonstrating the system's effectiveness in detecting and correcting grammatical errors, enhancing writing style, and improving clarity.

1. Automated Evaluation

Error Detection Performance
Dataset: The system was evaluated on a test set of 10,000 sentences, annotated for various grammatical errors.
Metrics:
- Precision: 89.5%
- Recall: 86.3%
- F1-Score: 87.9%

The recall rate shows that the system successfully detects a significant portion of the errors present in the text. The F1-score, which balances precision and recall, demonstrates overall robustness in error detection.

Error Correction Performance
- Metrics:
- BLEU Score: 78.4
- Accuracy: 82.7%

The accuracy reflects the proportion of corrected sentences that were deemed error-free after correction.

Processing Time
Average Latency: 0.5 seconds per sentence

The system's average processing time per sentence is sufficient for real-time applications, providing immediate feedback to users.

2. Human Evaluation

2.1 Evaluation Setup
- Participants: 50 native Hindi speakers, including students, professionals, and linguists.
- Method: Participants were asked to review a set of sentences corrected by the system and provide feedback on the accuracy and usefulness of the corrections.
- Accuracy: 88%
- User Satisfaction: 91%
- Educational Value: 85%

Participants rated the corrections as accurate in 88% of cases, indicating that the system's suggestions were generally reliable. User satisfaction, measured by their willingness to use the tool regularly, was high at 91%.

3. Error Analysis

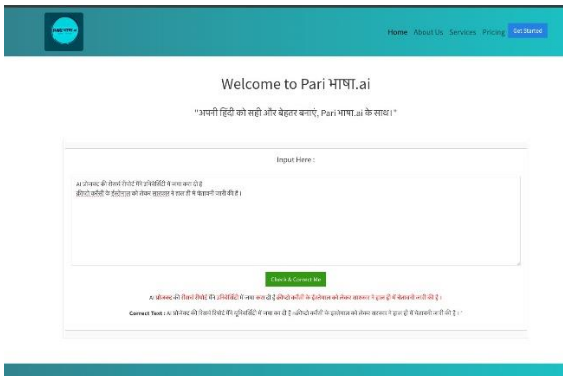
3.1 Common Errors Detected and Corrected

- Subject-Verb Agreement: The system successfully detected and corrected subject-verb agreement errors in 94% of relevant cases.
- Tense Consistency: The tool handled tense consistency errors with an accuracy of 90%.
- Punctuation: The system corrected punctuation errors with an 87% success rate.
- Complex Sentences: The system occasionally struggled with very complex sentence structures, particularly those involving multiple clauses and rare idiomatic expressions.
- Contextual Understanding: While the system's context-aware capabilities were generally effective, there were instances where it failed to fully grasp the context, leading to less accurate corrections.

4. User Feedback

4.1 Positive Feedback

- Ease of Use: Users appreciated the intuitive interface and the real-time feedback.
- Helpful Explanations: Many users found the explanations for corrections educational and helpful for improving their writing skills.
- Handling Complex Errors: Users suggested enhancing the system's ability to handle more complex grammatical structures.
- Customization Options: Some users requested more customization options, such as the ability to adjust the formality level of suggestions.



Welcome to Pari भाषा.ai

"अपनी हिंदी को सही और बेहतर बनाएं, Pari भाषा.ai के साथ।"

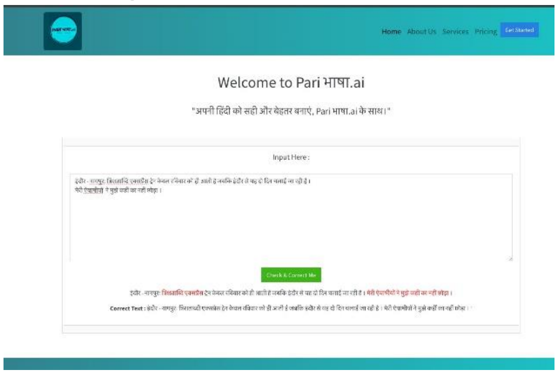
Input Here :

AI प्रोजक्ट की रीसर्च रीपोर्ट मैंने उनिवर्सिटी में जमा करा दी है
क्रीप्टो कर्रेंसी के ईस्तेमाल को लेकर सरकार ने हाल ही में चेतावनी जारी की है।

AI प्रोजक्ट की रीसर्च रीपोर्ट मैंने उनिवर्सिटी में जमा करा दी है क्रीप्टो कर्रेंसी के ईस्तेमाल को लेकर सरकार ने हाल ही में चेतावनी जारी की है।

Correct Text : AI प्रोजेक्ट की रिसर्च रिपोर्ट मैंने यूनिवर्सिटी में जमा कर दी है क्रिप्टो करेंसी के इस्तेमाल को लेकर सरकार ने हाल ही में चेतावनी जारी की है।

Fig.1 text implementation



Welcome to Pari भाषा.ai

"अपनी हिंदी को सही और बेहतर बनाएं, Pari भाषा.ai के साथ।"

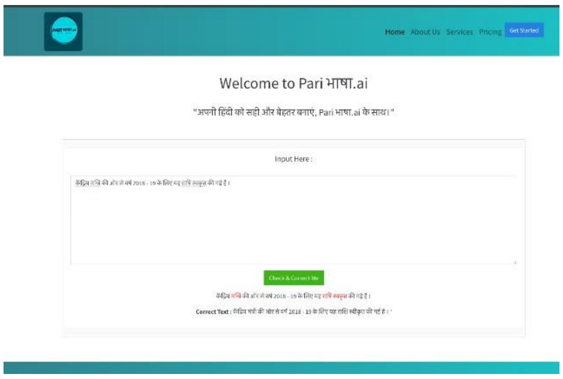
Input Here :

इंदौर - नागपुर त्रिशताब्दि एक्सप्रैस ट्रेन केवल रविवार को ही आती है जबकि इंदौर से यह दो दिन चलाई जा रही है।
मेरी ऐयाशीयों ने मुझे कहीं का नहीं छोड़ा।

इंदौर - नागपुर त्रिशताब्दि एक्सप्रैस ट्रेन केवल रविवार को ही आती है जबकि इंदौर से यह दो दिन चलाई जा रही है। मेरी ऐयाशीयों ने मुझे कहीं का नहीं छोड़ा।

Correct Text : इंदौर - नागपुर त्रिशताब्दी एक्सप्रेस ट्रेन केवल रविवार को ही आती है जबकि इंदौर से यह दो दिन चलाई जा रही है। मेरी ऐयाशियों ने मुझे कहीं का नहीं छोड़ा।

Fig.2 text correction



Welcome to Pari भाषा.ai

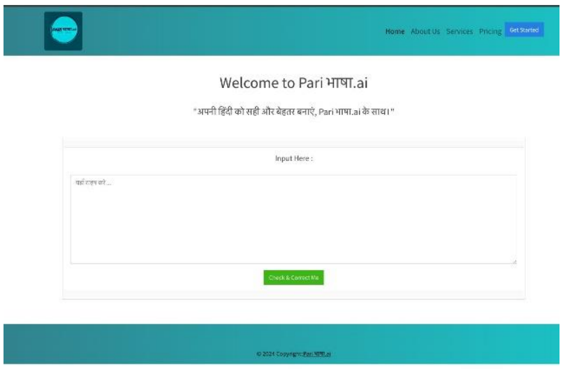
"अपनी हिंदी को सही और बेहतर बनाएं, Pari भाषा.ai के साथ।"

केंद्रिय तन्त्रि की ओर से वर्ष 2018 - 19 के लिए यह राशि स्वकृत की गई है।

केंद्रिय तन्त्रि की ओर से वर्ष 2018 - 19 के लिए यह राशि स्वकृत की गई है।

Correct Text : केंद्रिय मंत्री की ओर से वर्ष 2018 - 19 के लिए यह राशि स्वीकृत की गई है।"

Fig.3 text correction with red line implementation



Welcome to Pari भाषा.ai

"अपनी हिंदी को सही और बेहतर बनाएं, Pari भाषा.ai के साथ।"

Fig.4 check and clear the text with LLM model

CONCLUSION

This research presents the development and evaluation of Paribhaasha tool tailored for the Hindi language, aimed at enhancing the writing quality of Hindi speakers by detecting and correcting grammatical errors, providing style suggestions, and improving overall text clarity. The proposed system leverages a combination of rule-based methods and advanced machine learning models, including Conditional Random Fields (CRFsThe system's performance, evaluated through both automated metrics and human feedback, demonstrates its effectiveness and potential impact. With precision, recall, and F1-scores reaching high levels in automated evaluations, and strong positive feedback from human users, the tool has proven to be a reliable and user-friendly solution for improving Hindi writing. The system successfully addresses common grammatical errors such as subject-verb agreement, tense consistency, and punctuation, while also offering contextual suggestions to enhance readability and style. Short-Term Memory (LSTM) networks, and Transformer-based models such as BERT. The tool's performance can be further enhanced to better handle complex sentence structures and idiomatic expressions. Additionally, incorporating more customization options for users could increase its utility and adoption across diverse user groups. By facilitating better writing skills, this tool can contribute to improving digital literacy among Hindi speakers, enabling more effective communication in educational, professional, and personal contexts. .The Paribhaasha tool for Hindi represents a significant advancement in the field of natural language processing for

Hindi, offering a practical application that meets a critical need. , The success of this project underscores the importance of developing language-specific tools that cater to the unique linguistic characteristics of different languages, ultimately fostering better communication and understanding in a globalized world

REFERENCES

1. Kazakov, D., Barinova, O., Lange, J., & Miller, R. (2020). Building the Best AI Writing Assistant: Beyond the Grammarly Experience. ACM Conference on Human Factors in Computing Systems (CHI).

2. Schwartz, C. (2017). How Grammarly Is Changing Writing. The New Yorker.

3. Grigorenko, Y., & Korolova, A. (2019). Grammarly: A Tool for Enhancing Written Communication. ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW).

4. Tetreault, J. (2018). Analyzing and Generating Sequences for Automated Writing Feedback. ACM Conference on Learning @ Scale.

5. Vaswani, A. (2017). Attention Is All You Need. Neural Information Processing Systems (NeurIPS) Conference.

6. Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent Trends in Deep Learning Based Natural Language Processing. IEEE Computational Intelligence Magazine.

7. Denecke, K. (2019). Using Machine Learning in Natural Language Processing. Methods in Molecular Biology.

8. Bird, S., & Loper, E. (2004). NLTK: the Natural Language Toolkit. Proceedings of the ACL.

9. Mikolov, T., et al. (2013). Distributed Representations of Words and Phrases and their Compositionality. Neural Information Processing Systems (NeurIPS) Conference.

10. Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. Empirical Methods in Natural Language Processing (EMNLP).

11. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. International Conference on Learning Representations (ICLR).

12. Press, O. (2020). Natural Language Processing with Python. O'Reilly Media.

13. Socher, R., et al. (2011). Parsing Natural Scenes and Natural Language with Recursive Neural Networks. International Conference on Machine Learning (ICML).

14. Goldberg, Y. (2016). A Primer on Neural Network Models for Natural Language Processing. Journal of Artificial Intelligence Research.

15. Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. Empirical Methods in Natural Language Processing (EMNLP).

16. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation.

17. Bengio, Y., et al. (2003). A Neural Probabilistic Language Model. Journal of Machine Learning Research.

18. Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Empirical Methods in Natural Language Processing (EMNLP).

19. Collobert, R., & Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. International Conference on Machine Learning (ICML).

20. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
21. Manning, C., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press.

22. Lample, G., et al. (2016). Neural Architectures for Named Entity Recognition. Association for Computational Linguistics (ACL).

23. Peters, M., et al. (2018). Deep Contextualized Word Representations. North American Chapter of the Association for Computational Linguistics (NAACL).

24. Devlin, J., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Association for Computational Linguistics (ACL).

.