INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 07, July 2024, pp: 847-850

e-ISSN : 2583-1062

Impact Factor: 5.725

AUTOCORRECTION MODELS USING NATURAL LANGUAGE PROCESSING

Aruna Mrutyunjay Badiger[1], Shashank N N[2], Ankit Kumar[3], Abhishek Shetty B[4], Kulkarni Varsha[5]

[1,2,3,4]Third Year Students, Dept. of CSE, Sri Venkateshwara College of Engineering, Bengaluru, India.

[5]Assistant Professor, Dept. of CSE, Sri Venkateshwara College of Engineering, Bengaluru, India.

DOI: https://www.doi.org/10.58257/IJPREMS35481

ABSTRACT

This paper presents a comparative study of three prevalent approaches; N-Gram, Hidden Markov Model (HMM), and Rule-Based system in autocorrection systems for Natural Language Processing (NLP). Autocorrection systems are essential tools in digital communication platforms for rectifying typing errors and enhancing user experience. The N-Gram model employs statistical probabilities of word sequences to predict and rectify errors, exhibiting proficiency in local context comprehension but encountering challenges in handling out-of-vocabulary terms and capturing extensive dependencies. In contrast, the Hidden Markov Model excels in modeling sequential data and dependencies between observable and latent states, yet faces obstacles in parameter estimation and scalability. Furthermore, rule-based approaches leverage predefined linguistic rules and patterns for error detection and correction, offering transparency and customization flexibility, but struggling with intricate linguistic phenomena and language variations. Through empirical evaluations on standard datasets, this research compares the performance of these approaches in accuracy, computational efficiency, and robustness across diverse error types and language domains. Additionally, it discusses hybrid methodologies amalgamating the strengths of multiple techniques to enhance autocorrection performance. The findings contribute insights into the strengths and limitations of each approach, guiding future research and advancements in autocorrection systems for NLP applications.

1. INTRODUCTION

Autocorrection in Natural language processing is a feature that aims to predict and correct spelling errors while typing or writing. It is crucial in making tasks like composing paragraphs, reports, and articles more convenient. In today's digital landscape, many websites and social media platforms employ autocorrection to enhance user-friendliness. Autocorrection is firmly rooted in NLP. NLP is a field of artificial intelligence focusing on understanding and processing human language. It enables machines to interact with and generate natural language.

There are two main purposes of Autocorrection: 1. Correcting Spelling Errors where autocorrect identifies and rectifies misspelled words, ensuring accurate communication. 2. By enhancing User Experience where it automatically suggests corrections, streamlines text input, and reduces manual effort.

Autocorrection is a feature commonly found in text editors, messaging apps, and search engines. Its primary purpose is to predict and correct spelling errors as users type or write. By automatically suggesting the correct spelling of words, autocorrection enhances user experience and ensures accurate communication. The foundation of autocorrection relies on principles from Natural Language Processing (NLP) where NLP is a field of artificial intelligence focusing on understanding and processing human language.

Applications of Autocorrection:

- Auto Spelling Correction: Identifying and rectifying misspelled words.
- Sentiment Analysis: Analysing text to understand emotions expressed.
- Fake News Detection: Identifying misleading or fabricated information.
- Machine Translation: Enabling communication across languages.
- Question and Answering (Q&A): Extracting relevant answers from text.
- Chatbots: Engaging in natural conversations with users.

Current and Future Progress of NLP

Natural Language Processing has garnered significant attention for its ability to computationally represent and analyze human language. Researchers actively explore various linguistic phenomena, aiming to bridge the gap between language and computation.

Syntactic Phenomena focus on sentence structure and word order. These studies consider grammatical classes of words rather than their meaning. For instance, researchers develop discriminative models to score parses and explore coarse-to-fine efficient approximate parsing techniques. Dependency grammar also plays a crucial role in understanding

sentence structure. In the realm of Machine Translation, researchers work on improving models and algorithms. Challenges arise in low-resource languages and those with complex morphology. Efficient translation across diverse linguistic contexts remains a key goal.

Semantic Phenomena relate to sentence meaning, independent of context. Researchers delve into areas like sentiment analysis, which discerns emotions expressed in text. Summarization condenses lengthy content, while information extraction identifies relevant details. Slot-filling populates predefined slots, and discourse analysis studies sentence connections within larger contexts. Additionally, textual entailment determines logical relationships between sentences.

Pragmatic Phenomena (Speech) bridges sentence meaning and context. Linguistic and non-linguistic contexts play vital roles. Researchers model language syntax and semantics, explore acoustics, and address pronunciation challenges.

As speech recognition and information retrieval transition from research to commercial applications, vast amounts of text and speech data become available. Formalizing insights, mathematical formalism, algorithm development, and real-world testing contribute to advancing NLP. The study of language encompasses both discrete knowledge (what is possible) and continuous knowledge (what is likely). NLP continues to evolve, connecting human communication with computational understanding.

2. AUTOCORRECTION MODELS IN NLP

There are many different models for autocorrection by using natural language processing. For this comparative study, we considered three models: 1. Hidden Markov Models (HMMs) 2. N-Gram Language Models 3. Rule-Based System.

2.1 Hidden Markov Models (HMMs) in Autocorrection:

Hidden Markov Model (HMM) is a statistical model used for sequential data analysis. It is called “hidden” because some variables within the model are not directly observable, instead, they are inferred from the observable data. The following are the steps to Implement an HMM for Autocorrection:

- Define State Space and Observation Space: The state space represents all possible hidden states, while the observation space represents all possible observations.
- Initial State Distribution: Define the probability distribution over the initial state.
- State Transition Probabilities: Specify the probabilities of transitioning from one state to another. This forms the transition matrix.
- Observation Likelihoods: Determine the probabilities of generating each observation from each state. This forms the emission matrix.
- Model Training: Estimate the parameters (transition probabilities and observation likelihoods) using algorithms like the Baum-Welch algorithm or the forward-backward algorithm

2.2 N-Gram Language Models in Autocorrection

N-gram language Models play a crucial role in predicting the probability of a word based on the preceding words in a given sequence. These models are widely used in various NLP applications, including autocorrection, speech recognition, and text generation. An N-gram is a contiguous sequence of N items (such as letters, words, or base pairs) from a given sample of text or speech. N-grams are typically collected from a large text corpus. An N-gram language model predicts the probability of a given N-gram occurring within any sequence of words in the language. For example:

Unigram: (“This”, “article”, “is”, “on”, “NLP”)

Bigram: (“This article”, “article is”, “is on”, “on NLP”)

To find the next word in a sentence, we calculate the conditional probability p(w|h), where w is the candidate for the next word, and h represents the previous words. The chain rule of probability guides us: For unigrams: p(w) and For bigrams: p(w|h) = p(w|h1)

In Generalized formula using Markov assumptions: For unigrams: p(w), For bigrams: p(w|h) = p(w|h1) and For trigrams and beyond: p(w|h) = p(w|h1, h2, …, h[N-1])

2.3 Rule-Based System Models in Autocorrection:

Rule-based systems are one of the oldest and most straightforward approaches in NLP. In these systems, predefined linguistic rules are used to analyse and process textual data. Rule-based approaches involve applying specific sets of rules or patterns to capture structures, extract information, or perform tasks such as text classification. The key Concepts in the Rule-based System are:

1. Rule Creation: Based on the desired tasks, domain-specific linguistic rules are created. These rules can include:

- Grammar Rules: Defining sentence structures, verb-noun agreements, etc.
- Syntax Patterns: Identifying specific word sequences (e.g., subject-verb-object).
- Semantic Rules: Capturing meaning (e.g., identifying entities, and relationships).

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 07, July 2024, pp: 847-850

e-ISSN : 2583-1062

Impact Factor: 5.725

- Regular Expressions: Powerful tools for pattern matching.

2. Rule Application: The predefined rules are applied to input data (text) to capture matched patterns. For example, identifying verb phrases, noun phrases, or specific named entities.
3. Rule Processing: The text data is processed according to the results of the matched rules. This processing can involve an extraction of relevant information, making decisions based on specific patterns, and then correcting spelling errors (as in autocorrection).
4. Rule Refinement: Created rules are iteratively refined through repetitive processing. Feedback from previous iterations helps modify and update rules as needed. Refinement aims to improve accuracy and performance.
5. Libraries for Rule-Based Approaches: Several libraries can be used for rule-based NLP are

- Spacy: A powerful library for advanced NLP tasks. It includes a rule-matching engine called the Matcher, which works over tokens, entities, and phrases.
- fast.ai: Useful for rule-based approaches.
- NLTK (Natural language Toolkit): Although not commonly preferred nowadays, NLTK can still be used for rule-based tasks.

Quality Measures for Autocorrection Models:

To check the quality measures of the autocorrection models, we have applied the same input to all models. Some of the measures that have been considered are as follows:

For Rule Based & HMM Model

Accuracy: This is the proportion of correctly predicted states or observations out of the total number of predictions. It measures how often the model agrees with the true data.

Precision: This is the proportion of correctly predicted positive states or observations out of the total number of positive predictions. It measures how reliable the model is when it predicts a positive outcome.

Recall: This is the proportion of correctly predicted positive states or observations out of the total number of true positive outcomes. It measures how well the model can capture the positive cases in the data.

F1 score: This is the harmonic mean of precision and recall. It measures the balance between precision and recall.

For N-Gram Model

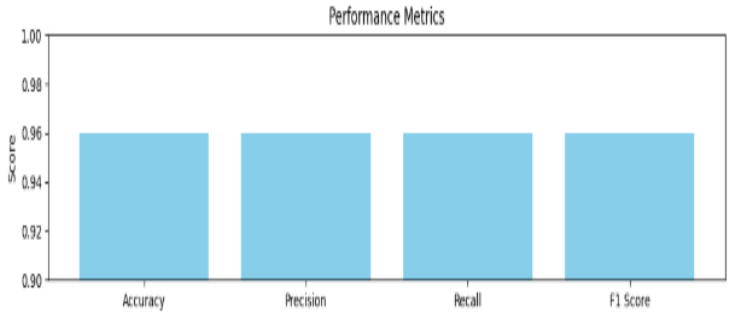
Perplexity: This is the inverse probability of the test set, normalized by the number of words. It measures how surprised the model is by the test data. A lower perplexity means a better fit of the model to the data.

Entropy: This is the average amount of information in each test set word. It measures the uncertainty of the model about the next word. A higher entropy means a more diverse vocabulary and a more complex language.

Log-likelihood: This is the logarithm of the probability of the test set given the model. It measures how likely the model is to generate the test data. A higher log-likelihood means a more probable model.

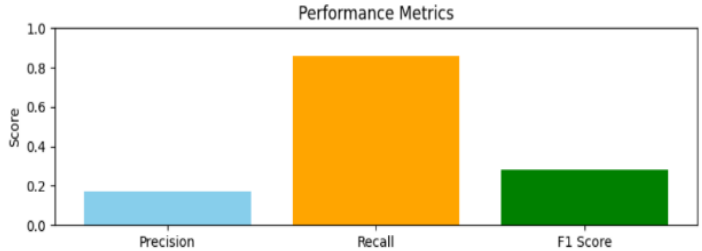
3. RESULTS

HMM model



The Hidden Markov Model (HMM) demonstrates remarkable accuracy, precision, recall, and F1 score in predicting hidden states from observed data. This model is more accurate than both N-gram and rule-based models.

Rule-based models



INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

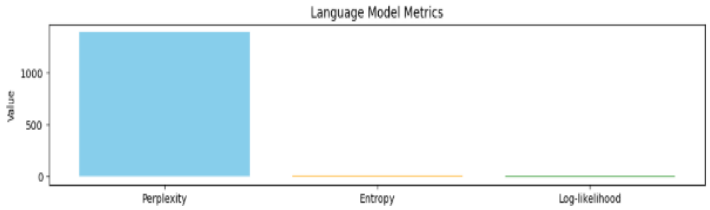
Vol. 04, Issue 07, July 2024, pp: 847-850

e-ISSN : 2583-1062

Impact Factor: 5.725

The performance evaluation of the rules-based autocorrection algorithm in NLP reveals suboptimal results for spelling error correction. It exhibits low precision, indicating a high rate of false positives. It demonstrates high recall, implying that it captures most of the relevant errors but also generates false positives. The F1 score, which balances precision and recall, is also subpar. To enhance system performance.

N-gram models



The N-gram language model, while widely used, exhibits limitations in predicting the subsequent word in a given test sentence. It is characterized by elevated perplexity and high entropy, resulting in a suboptimal log-likelihood.

4. CONCLUSION

Autocorrection, a pivotal feature, aims to predict and rectify spelling errors during text input or composition. Its significance lies in streamlining tasks such as drafting paragraphs, generating reports, and crafting articles. In the contemporary digital landscape, numerous websites and social media platforms leverage autocorrection to enhance user experience and text quality. NLP, as an AI discipline, focuses on deciphering and manipulating human language, enabling machines to interact with and generate coherent natural language expressions. Hidden Markov Models (HMMs), n-gram models, and rule-based systems. HMMs, rooted in probabilistic modeling, excel at capturing sequential dependencies within language data. However, their limitations lie in the need for substantial training data and the Markovian assumption. N-gram models offer simplicity and efficiency but struggle with handling long-range dependencies. Rule-based systems, while intuitive, face challenges in handling irregularities and domain-specific jargon. Moving forward, we recommend exploring hybrid models, personalized customization, and pre-trained contextual embeddings to enhance autocorrection performance.

5. REFERENCES

[1] Adam Pauls Dan Klein, 2011 Faster and Smaller N-Gram Language Models

[2] Daniel Jurafsky & James H. Martin 2024 Speech and Language Processing, N-gram Language Models

[3] Stephanie Seneff, Chao Wang, and Timothy J. Hazen 2003 Automatic Induction of N-Gram Language Models from a Natural Language Grammar

[4] VOLUME 6, ISSUE 3 (March 2016) (ISSN 2249-3905) International Journal of Research in Engineering and Applied Sciences (IMPACT FACTOR – 6.573)

[5] Prakash M Nadkarni, Lucila Ohno-Machado, Wendy W Chapman 6 July 2011 Natural language processing: an introduction