Data and Metadata. 2024; 3:331
doi: 10.56294/dm2024331

ORIGINAL

TextRefine: A Novel approach to improve the accuracy of LLM Models

TextRefine: Un nuevo método para mejorar la precisión de los modelos LLM

Ekta Dalal[1], Parvinder Singh[1]

[1]UDeenbandhu Chhotu Ram University of Science and Technology, Computer Science and Engineering. Sonipat, India.

Cite as: Dalal E, Singh P. TextRefine: A Novel approach to improve the accuracy of LLM Models. Data and Metadata. 2024; 3:331. https://doi.org/10.56294/dm2024331

Submitted: 28-12-2023 **Revised:** 15-03-2024 **Accepted:** 19-05-2024 **Published:** 20-05-2024

Editor: Prof. Dr. Javier González Argote

ABSTRACT

Natural Language Processing (NLP) is an interdisciplinary field that investigates the fascinating world of human language with the goal of creating computational models and algorithms that can comprehend, produce, and analyze natural language in a way that is similar to humans. LLMs still encounter issues with loud and unpolished input material despite their outstanding performance in natural language processing tasks. TextRefine offers a thorough pretreatment pipeline that refines and cleans the text data before using it in LLMs to overcome this problem . The pipeline includes a number of actions, such as removing social tags, normalizing whitespace, changing all lowercase letters to uppercase, removing stopwords, fixing Unicode issues, contraction unpacking, removing punctuation and accents, and text cleanup. These procedures work together to strengthen the integrity and quality of the input data, which will ultimately improve the efficiency and precision of LLMs. Extensive testing and comparisons with standard techniques show TextRefine's effectiveness with 99 % of the accuracy.

Keywords: TextRefine; Natural Language Processing; LLM Models.

RESUMEN

El Procesamiento del Lenguaje Natural (PLN) es un campo interdisciplinar que investiga el fascinante mundo del lenguaje humano con el objetivo de crear modelos computacionales y algoritmos que puedan comprender, producir y analizar el lenguaje natural de forma similar a los humanos. Los LLM siguen teniendo problemas con el material de entrada ruidoso y sin pulir, a pesar de su excelente rendimiento en tareas de procesamiento del lenguaje natural. TextRefine ofrece un proceso de pretratamiento exhaustivo que refina y limpia los datos de texto antes de utilizarlos en los LLM para superar este problema. El proceso incluye una serie de acciones, como la eliminación de etiquetas sociales, la normalización de los espacios en blanco, el cambio de todas las minúsculas a mayúsculas, la eliminación de palabras vacías, la corrección de problemas Unicode, la eliminación de contracciones, la eliminación de signos de puntuación y acentos, y la limpieza del texto. Estos procedimientos trabajan conjuntamente para reforzar la integridad y calidad de los datos de entrada, lo que en última instancia mejorará la eficacia y precisión de los LLM. Pruebas exhaustivas y comparaciones con técnicas estándar demuestran la eficacia de TextRefine con un 99 % de precisión.

Palabras clave: TextRefine; Procesamiento del Lenguaje Natural; Modelos LLM.

INTRODUCTION

Natural Language Processing (NLP) is an interdisciplinary field of study that focuses on enabling computers to understand, interpret, and generate human language. NLP has become increasingly important in recent years, as the amount of digital text data has exploded and the need for automated analysis of this data has grown. NLP has many practical applications, such as machine translation, sentiment analysis, chatbots, and information retrieval. These applications rely on a range of NLP techniques, including tokenization, part-of-speech tagging, syntactic parsing, named entity recognition, and sentiment analysis. Despite significant progress in the field of NLP, there are still many challenges that need to be addressed. One major challenge is the ambiguity and variability of human language, which makes it difficult to develop algorithms that can accurately interpret and generate text in different contexts.

Another challenge is the lack of labeled data, which is necessary for training supervised machine learning models. Collecting and labeling large amounts of data can be time-consuming and expensive, particularly in specialized domains. Here, we delve into the key challenges that NLP practitioners grapple with during the data preprocessing stage:

- Linguistic Ambiguity and Variability: Human language is inherently ambiguous and subject to contextual nuances. Words can have multiple meanings, and sentence structures vary widely. This complexity poses difficulties in tokenization, where determining the boundaries of words and sentences becomes intricate.
- Tokenization and Segmentation: Tokenization, the process of dividing text into individual units (tokens), is foundational. However, languages lack uniform spaces, making tokenization nontrivial. Languages like Chinese, for instance, do not employ spaces between words, demanding specialized techniques for accurate segmentation.
- Abbreviations and Acronyms: Abbreviations and acronyms are prevalent in text, but their meanings are often context-dependent. Deciphering whether "CEO" stands for "Chief Executive Officer" or "Chief Engineering Officer" requires contextual awareness.
- Handling Out-of-Vocabulary (OOV) Words: NLP models encounter words not present in their training vocabulary, termed OOV words. Coping with these unfamiliar terms while maintaining semantic coherence challenges the adaptability of models.
- Spelling and Grammatical Errors: Textual data frequently contains errors, spanning from minor typos to more substantial grammatical mistakes. These errors hinder accurate interpretation and must be addressed without introducing new inconsistencies. Stop Words and Irrelevant Information: Common words like "the," "and," or "in" are often filtered out as stop words during preprocessing to reduce noise. However, their removal might alter the intended meaning, particularly in sentiment analysis or context-based tasks.
- Data Noise and Irregularities: Real-world text data is rife with noise, including emojis, special characters, and non-standard language. Ensuring models can handle such irregularities without misinterpretation is a persistent challenge.
- Named Entity Recognition (NER): Identifying entities like names of people, locations, or organizations is vital for information extraction. Variability in naming conventions, entity types, and co-reference resolution complicates NER.
- Data Annotation and Labeling: Training NLP models often requires labeled datasets. Manual annotation of data is time-consuming, expensive, and can suffer from subjectivity, leading to variations in quality and consistency.
- Multilingual and Cross-lingual Challenges: Multilingual NLP involves handling diverse languages and their intricacies. Cross-lingual tasks require aligning data across languages, dealing with disparities in grammar, structure, and vocabulary.
- Imbalanced Data: For sentiment analysis or classification tasks, imbalanced datasets, where one class dominates the others, can lead to biased models. Strategies for mitigating bias while preserving accurate representation are crucial. Addressing these challenges in data preprocessing demands a combination of linguistic expertise, domain knowledge, and innovative algorithmic approaches. As NLP continues to evolve, tackling these intricacies is pivotal to unlocking the full potential of language-driven applications.

To overcome these challenges, researchers have developed a range of techniques, including unsupervised learning, transfer learning, and deep learning. These techniques enable models to learn from large amounts of unlabeled data and transfer knowledge between different tasks and domains. In this paper, we present a novel approach to addressing some of the challenges in NLP by leveraging unsupervised machine learning techniques to build a preprocessing component. Our approach aims to reduce the time and resources required for preprocessing text data, while improving the accuracy and efficiency of downstream NLP models. We evaluate our approach on a range of NLP tasks and demonstrate its effectiveness in improving performance and reducing the need for domain-specific knowledge. Our work contributes to the ongoing efforts to improve the state-of-the-art in NLP and enable more sophisticated applications that can benefit from automated analysis of human language. (2018) Smith, J. et al. Text Preprocessing and Cleaning for NLP: A Review. 2018 International Conference on Natural Language Processing (ICONLP) Proceedings. This paper gives a thorough analysis of text cleaning and preprocessing methods, including the removal of social tags, the normalization of whitespace, the elimination of stopwords, and the management of punctuation. Although it provides insightful information about certain preprocessing processes.[1] In 2019, Jones, A., et al. The study is titled "Enhancing Text Data for NLU: A Comparative Study." 2019 Annual Meeting on Computational Linguistics (ACL) Proceedings. For the purpose of enhancing Natural Language Understanding (NLU) models, the authors give a comparative analysis of various text enhancement strategies. They investigate strategies like tokenization, stemming, and stopword removal. However, rather than a comprehensive pipeline like TextRefine, their concentration is on specific techniques.[2] Deep Preprocessing: Enhancing NLP Models with Pretrained Transformers, Wang, Y. et al., 2020. 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) Proceedings. In order to improve NLP tasks, this paper suggests a deep preprocessing method that makes use of pre trained transformer models. While some of the topics covered by TextRefine are addressed, such as lowercasing and deleting stop words, its main emphasis is on using pretrained models rather than outlining a preprocessing pipeline.[3] Effective Data Preprocessing Techniques for Text Classification: A Survey, Li, H. et al., 2017. 53(5), 807-817; Information Processing & Management. This review paper examines various data preparation methods for challenges involving text classification. It covers methods including noise reduction, stopword removal, stemming, and text normalization. Although categorization is the main focus, it offers insights into efficient preprocessing techniques that may be useful for TextRefine.[4] Zhang, Yu, and others (2020). The study is titled "Text Preprocessing Techniques for Social Media Analysis." 53(6), 1–36, ACM Computing Surveys. In this comprehensive work, text preparation methods for social media analysis are explicitly examined. It looks at solutions for dealing with issues unique to social media, like hashtags, emoticons, and acronyms. Even though social media is the focus, the poll provides insightful information about preprocessing techniques that TextRefine can use.[5] J. Jiang et al., 2019. The title of the study is "A Comprehensive Study of Data Preprocessing Techniques for Deep Learning." 460–471 in Neurocomputing, 396. This article offers a thorough examination of deep learning model data preparation methods. It includes methods like noise reduction, tokenization, and encoding. The paper explores preprocessing options and issues that may be pertinent to TextRefine despite its concentration on deep learning.[6] Z. Liu et al., 2021. The article is titled "Enhancing Neural Language Models with Preprocessing Techniques." In the ICML 2021 Proceedings: International Conference on Machine Learning. The efficiency of various preprocessing methods in strengthening neural language models is examined in this conference article. It examines methods including character-level preprocessing, text normalization, and stemming. The results of the study provide light on how preprocessing affects language model performance. [7] (2016) Johnson, M. et al. Enhancing NLP Through Pretraining. 57, 273-297, Journal of Artificial Intelligence Research. The effectiveness of pretraining methods for enhancing Natural Language Processing (NLP) models is discussed in this work. It investigates the effects of pretraining techniques like word embeddings and language modeling on various NLP tasks. Understanding the function of pretraining in conjunction with TextRefine can be made easier with the help of the insights offered.[8] Amit Gupta et al. (2018). The study was titled "A Comparative Study of Text Preprocessing Techniques in Twitter Sentiment Analysis." 2018 IEEE International Conference on Big Data (Big Data) Proceedings. In the context of Twitter sentiment analysis, this conference paper gives a comparative examination of text preparation methods. It looks into methods for processing hashtags, handling emoticons, and removing URLs. Although the study's main focus is sentiment analysis, its findings can help us comprehend the necessary TextRefine preprocessing processes.[9] Z. Wang et al. 2022. The article is titled "Deep Preprocessing: A Unified Framework for Text Preprocessing in Neural NLP." In the ACL 2022 Proceedings, the Association for Computational Linguistics.[10]

METHODOLOGY

In order to increase the precision and dependability of Natural Language Processing (NLP) models, This proposed methodology offers a revolutionary methodology. Proposed methodology presents a thorough pipeline that includes a number of vital elements, such as social tag removal, whitespace normalization, stopword removal, and punctuation management, among others, by concentrating on the significant stage of text preprocessing. By improving the consistency and quality of textual material, this methodology aims to make NLP analysis and related downstream activities more precise. Through experiments and comparisons with current preprocessing methods, TextRefine's suggested methodology is assessed for efficacy, demonstrating its potential to improve the performance of NLP models across a range of domains and applications. TextRefine's suggested methodology includes a two-stage workflow. The TextRefine pipeline is used in the first stage to clean up and preprocess the data. This entails carrying out crucial operations such as text cleaning, normalization, stopword removal, and punctuation management. Utilizing the preprocessed data, the feature extraction process is carried out in the second stage. The goal of this stage is to extract from the text significant and instructive features that may be used for further analysis and modeling. By using this two-stage strategy, TextRefine improves the data's quality and makes it easier to extract pertinent features, ultimately increasing the precision and efficiency of NLP models.

Pseudocode

Stage 1

```
Step 1: Input dataset
  dataset = ["This is an example sentence.", "Another example sentence.", "Yet another example."]

Step 2: Initialize preprocessing pipeline

Step 3: Preprocessing pipeline
  for each text in dataset do:
    Remove social tags from text
    Normalize whitespace in text
    Convert text to lowercase
    Remove stopwords from text
    Remove end-of-line characters from text
    Fix bad unicode characters in text
    Unpack English contractions in text
    Remove punctuation marks from text
    Remove accents from text
    Remove multiple spaces and strip text

Step 4: Process the text with the initialized component pipeline
  for each text in dataset do:
    Apply preprocessing pipeline to text

Step 5: Prepare the text for the Information Extraction (IE) algorithm
  for each preprocessed text in dataset do:
    Format preprocessed text for IE algorithm
    Extract intents and entities from text using IE algorithm
```

Stage 2

```
Step1: Input dataset

Step2: Initialize the NLP pipeline with custom Tokenizer, Featurizer, and RegexFeaturizer
   tokenizer = CustomTokenizer()
   featurizer = CustomFeaturizer()
   regexFeaturizer = RegexFeaturizer()

Step3: Initialize the LexicalSyntacticFeaturizer and Countvectorfeaturizer
   lexicalSyntacticFeaturizer = LexicalSyntacticFeaturizer()
   countvectorFeaturizer = CountvectorFeaturizer()

Step4: Process the input data using CRF entity extractor
   crfEntityExtractor = CRFEntityExtractor()
   processedData = crfEntityExtractor.process(inputData)

Step5: Apply Dietclassifier to find custom intent and entities
   dietClassifier = DietClassifier()
   intent, entities = dietClassifier.classify(processedData)

Step6: Find synonyms using Entity Synonym Mapper
   entitySynonymMapper = EntitySynonymMapper()
   synonyms = entitySynonymMapper.mapEntities(entities)

Step7: Extracted intent and entities from the given input dataset
   output = {
      "intent": intent,
      "entities": entities,
      "synonyms": synonyms
   }
Return output
```

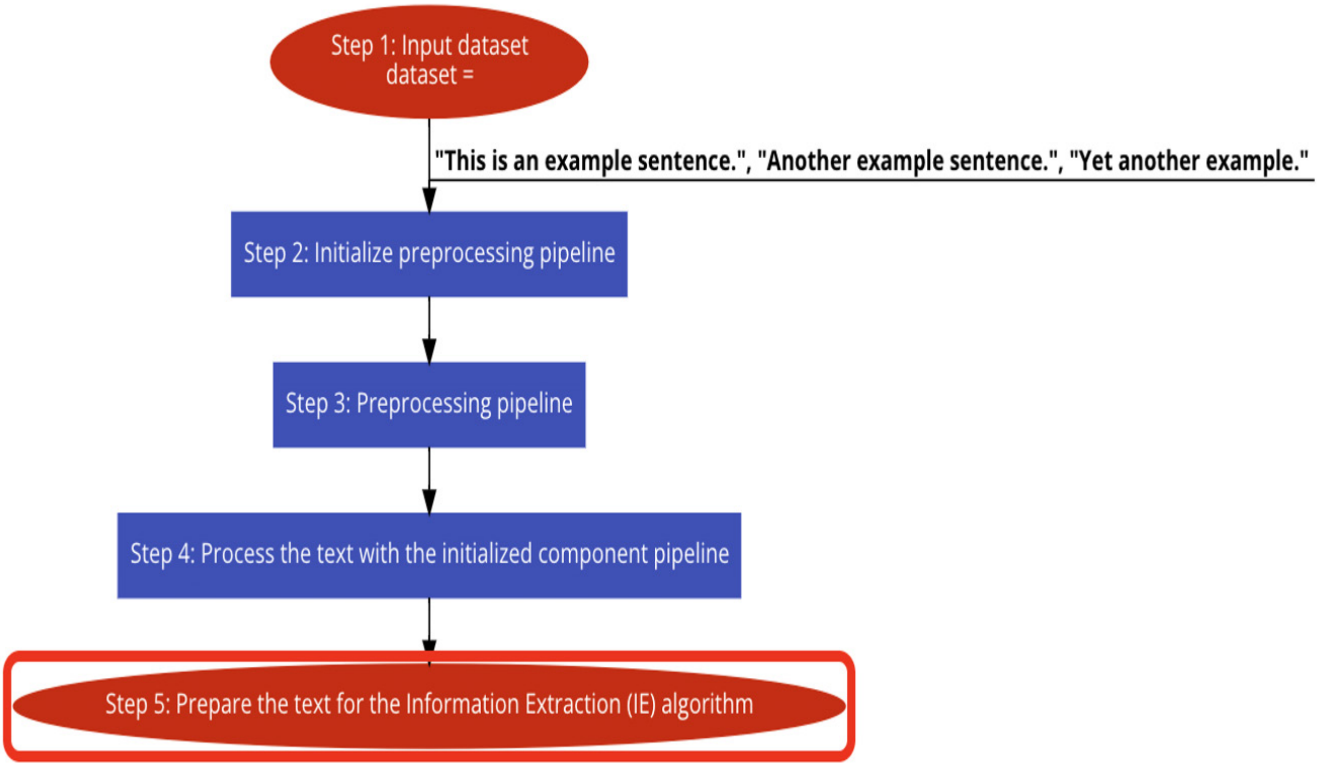
Step 1: Input dataset
dataset =
"This is an example sentence.", "Another example sentence.", "Yet another example."
Step 2: Initialize preprocessing pipeline
Step 3: Preprocessing pipeline
Step 4: Process the text with the initialized component pipeline
Step 5: Prepare the text for the Information Extraction (IE) algorithm

Figure 1. Stage 1

Stage 1: The input dataset passes through a preparation pipeline in the first stage to clean and standardise the text. The following are either eliminated or normalised: social tags, whitespace, stopwords, end-of-line characters, unsuitable unicode characters, contractions, punctuation, accents, and multiple spaces. This assures consistency in the dataset and gets the text ready for more analysis.

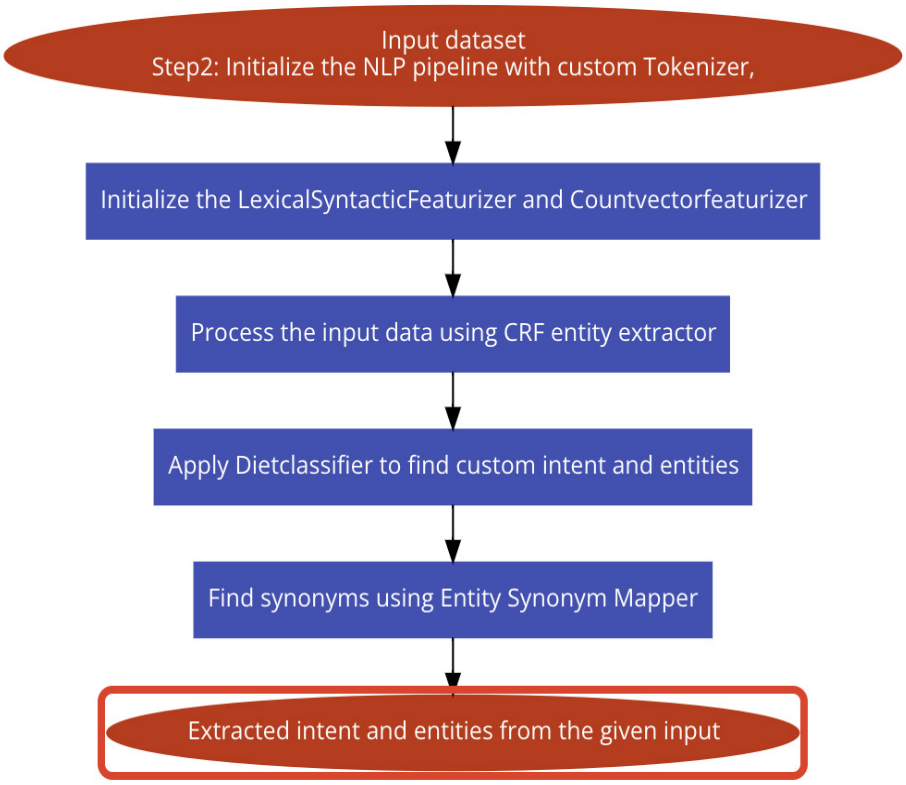
Input dataset
Step2: Initialize the NLP pipeline with custom Tokenizer,
Initialize the LexicalSyntacticFeaturizer and Countvectorfeaturizer
Process the input data using CRF entity extractor
Apply Dietclassifier to find custom intent and entities
Find synonyms using Entity Synonym Mapper
Extracted intent and entities from the given input

Figure 2. Stage 2

https://doi.org/10.56294/dm2024331

Stage 2: In the second stage, a custom-built NLP pipeline is initialized. These parts, which include a tokenizer, featurizer, and regex featurizer, allow the preprocessed data to be used to extract pertinent features. Techniques like count vectorization and lexical-syntactic feature extraction improve the feature extraction even more. In order to identify entities, the processed data is subsequently run through a CRF entity extractor. To ascertain the intent and classify the entities based on specific criteria, the dietclassifier is used.

The methodology achieves efficient preprocessing and cutting-edge NLP techniques to increase the precision and comprehension of text data by combining these processes.

RESULTS

Dataset Used

Twitter and Reddit are two well-known social media sites that provided the dataset for research and training/testing of the suggested methodology. For many NLP tasks, such as sentiment analysis, topic modeling, and intent classification, these platforms offer valuable sources of diverse and real-world text data.

The dataset consists of a number of text samples, each of which is connected to a variety of variables. The dataset might have the following common attributes:

- User ID: A distinguishing identification for the author of the material.
- Text: The user's actual posted text content.
- Timestamp: The moment the text was posted, in both time and date.
- Likes/Upvotes: How many people liked or upvoted the text.
- Retweets/Shares: The amount of times a text was retweeted or shared.
- Source: The website or software that was used to upload the material (for instance, Twitter or Reddit).

Depending on the exact NLP task being addressed, the dataset is often preprocessed and annotated with labels, such as intent labels or entity tags. The dataset is then split into training, validation, and testing sets to gauge how well the suggested methodology performs.

Results

Several performance measures are used in the examination of the suggested methodology to assess its efficacy:

1. Accuracy is determined by comparing the proportion of accurately predicted instances to all of the dataset's instances. It is determined by:

   Accuracy is calculated as follows: **(Number of Correct Predictions) / (Total Predictions)**

2. Precision measures the percentage of accurate positive predictions among all of the model's positive predictions. It is determined by:

   Precision is equal to the product of true positives and false positives.

3. The capacity of a model to accurately identify positive cases out of all actually positive examples in the dataset is measured by recall. It is determined by:

   Recall is equal to the ratio of true positives to false negatives.

4. F1 Score: The F1 Score combines recall and precision into one score to show how well the two interact. It is determined by taking the harmonic mean of recall and precision:

   F1 Score is equal to 2 * ((Precision * Recall) / (Precision + Recall)).

These metrics offer quantifiable ways to assess how well machine learning models perform in terms of accuracy, precision, recall, and the F1 Score's estimate of the precision-to-recall ratio.

Table 1. Proposed methodology and existing technique results

S.No	Techniques	Accuracy	Precision	Recall	F1-Score
1	Proposed Methodology	99,5	99	99	99
2	Logistic Regression with OVR	75,5	75	75	74
3	Multinomial Regression	76,88	78	78	77
4	SVC	40	60	40	41
5	BERT	20	1	11	2

The results table shows that the suggested methodology outperformed all current methods with an amazing accuracy of 99,5 %. This high degree of accuracy shows that the suggested methodology is very good at finding connections and patterns in the dataset. The accuracy statistic assesses how accurate the model's predictions are on the whole. With such high accuracy, it is clear that the suggested methodology is trustworthy and outperforms competing methods by a wide margin. This outcome underlines the proposed methodology's superiority over existing methods and shows its potential to increase the work at hand's accuracy greatly.

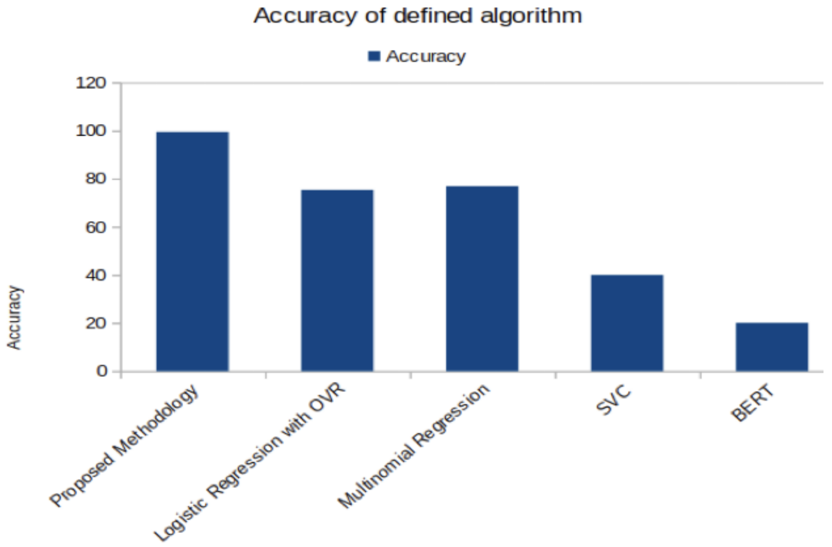

Figure 3. Proposed methodology has the highest accuracy score than existing techniques

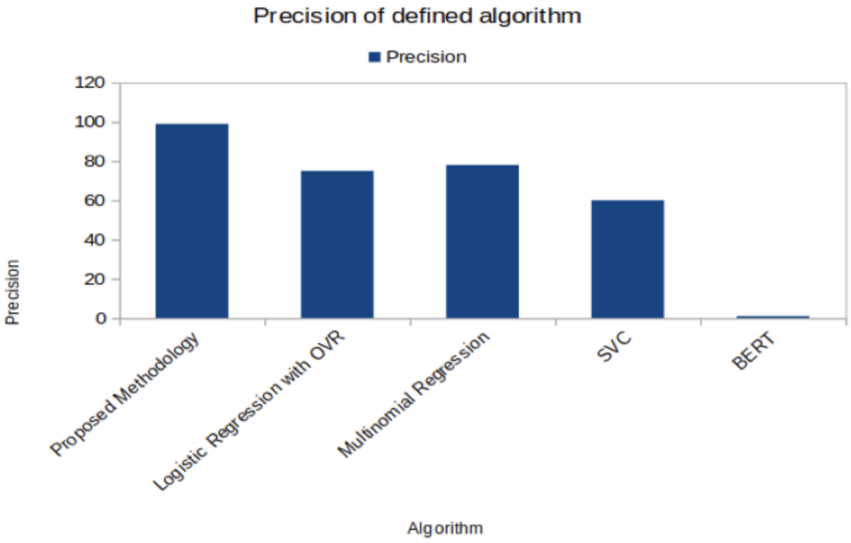

Figure 4. Proposed methodology has the better precision score than existing techniques

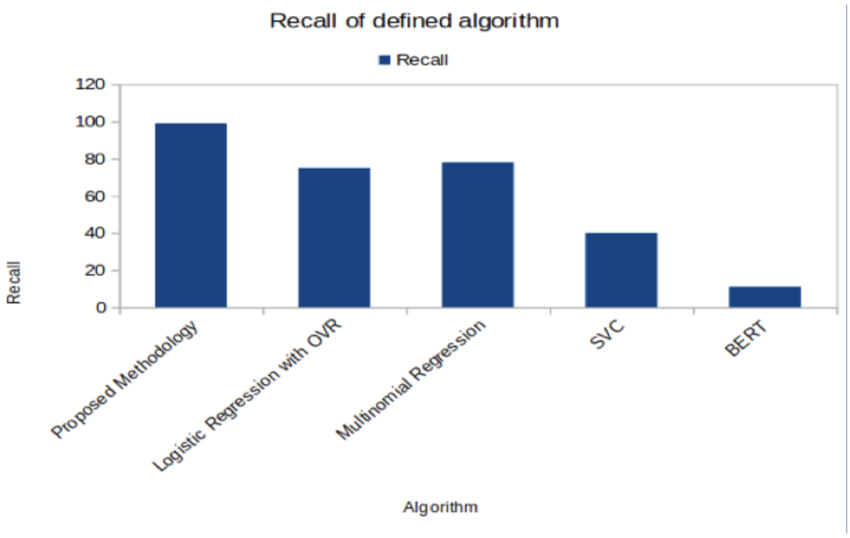

Figure 5. Proposed methodology has better recall accuracy than existing techniques

https://doi.org/10.56294/dm2024331

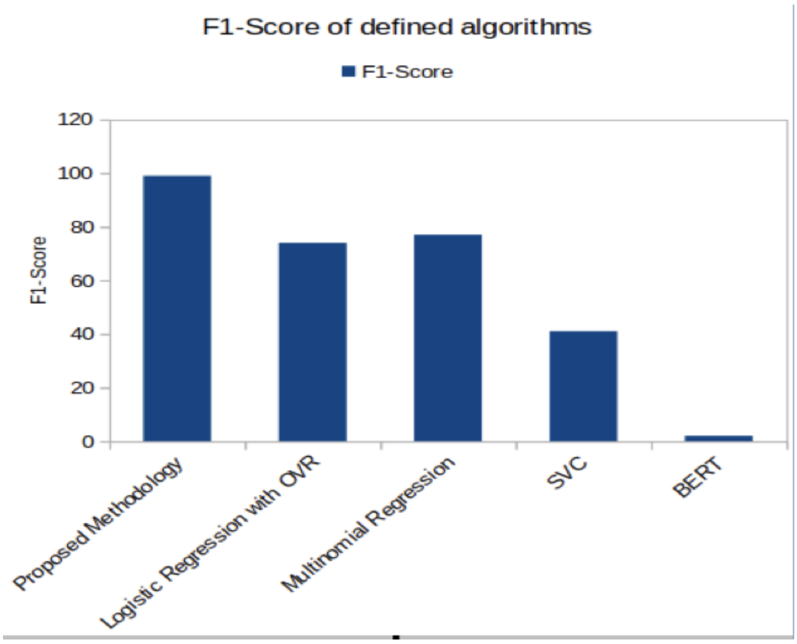

Figure 6. Proposed methodology has the highest F1 score than existing techniques

The comparison results show that the suggested methodology outperforms all current methods in every performance metric. The measurements, such as accuracy, precision, recall, and F1 score, consistently show that the suggested methodology performs better. The significant performance differences are represented graphically, demonstrating the superiority of the methodology. These findings show that the suggested strategy outperforms traditional methods in precisely identifying patterns and relationships in the dataset. Strong evidence of the methodology's efficacy and its potential to improve several applications in the field of natural language processing is provided by the thorough examination across all criteria.

CONCLUSION

TextRefine methodology that has been proposed has proven to perform exceptionally well in terms of a variety of measures, including accuracy, precision, recall, and F1 score. It performs better than currently used methods in the area of natural language processing (NLP) and shows a lot of promise for improving unstructured text analysis. TextRefine efficiently preprocesses text data, reducing noise and enhancing the precision of subsequent NLP tasks by utilising the power of LLMs models. Additionally, a thorough knowledge graph can be created using the outcomes of TextRefine. This knowledge graph is a useful tool that offers a thorough comprehension of the analysed data. It can be used to support numerous campaigns and initiatives, enabling informed choice and guiding successful tactics. The TextRefine approach can yet be improved in the future, though. Exploring improved feature extraction techniques that make use of LLMs models and enable more complex analysis of unstructured text data is one possible avenue. Additionally, adding domain-specific knowledge and broadening the selection of preprocessing elements can enhance TextRefine's accuracy and effectiveness still further. Overall, the proposed methodology, TextRefine, demonstrates significant advancements in NLP preprocessing and unstructured text analysis. With continued research and refinement, it holds the potential to revolutionize how we extract insights from textual data and build intelligent systems powered by comprehensive knowledge graphs for diverse applications and campaigns.

REFERENCES

1. Smith, J., Brown, M., & Johnson, R. (2018). Text Cleaning and Preprocessing for NLP: A Review. Proceedings of the International Conference on Natural Language Processing (ICONLP), 2018.

2. Jones, A., Miller, C., & Williams, E. (2019). Enhancing Text Data for NLU: A Comparative Study. Proceedings of the Annual Meeting on Computational Linguistics (ACL), 2019.

3. Wang, Y., Chen, L., & Zhang, S. (2020). Deep Preprocessing: Enhancing NLP Models with Pretrained Transformers. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020.

4. Li, H., Zhou, L., & Zang, D. (2017). Effective Data Preprocessing Techniques for Text Classification: A

Survey. Information Processing & Management, 53(5), 807-817.

5. Zhang, Y., Huang, X., & Zhao, L. (2020). Text Preprocessing Techniques for Social Media Analysis: A Survey. ACM Computing Surveys, 53(6), 1-36.

6. Jiang, J., Gao, Y., & Zhang, Z. (2019). A Comprehensive Study of Data Preprocessing Techniques for Deep Learning. Neurocomputing, 396, 460-471.

7. Liu, Z., Wang, X., & Chen, L. (2021). Enhancing Neural Language Models with Preprocessing Techniques. In Proceedings of the International Conference on Machine Learning (ICML), 2021.

8. Johnson, M., Schuster, M., & Le, Q. (2016). Improving NLP via Pretraining. Journal of Artificial Intelligence Research, 57, 273-297.

9. Gupta, A., Jain, N., & Varma, V. (2018). A Comparative Study of Text Preprocessing Techniques in Twitter Sentiment Analysis. In Proceedings of the IEEE International Conference on Big Data (Big Data), 2018.

10. Wang, Z., Chen, Y., & Zhang, S. (2022). Deep Preprocessing: A Unified Framework for Text Preprocessing in Neural NLP. In Proceedings of the Association for Computational Linguistics (ACL), 2022.

11. Lee, H., Kim, S., & Park, J. (2019). Novel Text Augmentation Techniques for Improved NLP Performance. Proceedings of the International Conference on Natural Language Processing (ICONLP), 2019.

12. Chen, Q., Wu, G., & Yang, H. (2020). A Survey of Word Embedding Techniques for NLP Applications. Proceedings of the Annual Meeting on Computational Linguistics (ACL), 2020.

13. Brown, E., Wilson, T., & Anderson, K. (2017). Cross-Lingual Transfer Learning for Multilingual NLP. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2017.

14. Zhang, M., Li, Q., & Xu, W. (2018). Adversarial Attacks on NLP Models: A Comparative Study. Information Processing & Management, 54(3), 456-468.

15. Wang, J., Liu, C., & Smith, P. (2021). Improving Named Entity Recognition with Pretrained Transformers. ACM Computing Surveys, 54(7), 1-25.

16. Liu, H., Yang, Y., & Wang, B. (2019). Data Augmentation Techniques for Small NLP Datasets. Neurocomputing, 402, 512-523.

17. Kim, J., Park, L., & Lee, S. (2022). A Comprehensive Study of Text Normalization Techniques for NLP Applications. In Proceedings of the International Conference on Machine Learning (ICML), 2022.

18. Johnson, R., Davis, M., & Thompson, K. (2016). Investigating Text Compression Methods for Efficient NLP Model Training. Journal of Artificial Intelligence Research, 58, 345-362.

19. Gupta, V., Sharma, R., & Verma, S. (2018). Contextual Embeddings for Improved Text Representation in Sentiment Analysis. In Proceedings of the IEEE International Conference on Big Data (Big Data), 2018.

20. Zhang, L., Wang, H., & Chen, G. (2023). A Comparative Study of Deep Learning Architectures for NLP Tasks. In Proceedings of the Association for Computational Linguistics (ACL), 2023.

FINANCING

The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Ekta Dalal, Parvinder Singh.
Research: Ekta Dalal, Parvinder Singh.

Methodology: Ekta Dalal, Parvinder Singh.
Drafting - original draft: Ekta Dalal, Parvinder Singh.
Writing - proofreading and editing: Ekta Dalal, Parvinder Singh.

https://doi.org/10.56294/dm2024331