Academic Journal of Natural Science
Journal Home: http://ajns.suaspress.org | CODEN: AJNSAE
Vol. 2, No. 2, 2025 | ISSN 3078-5170 (Print) | ISSN 3078-5189 (Online) | ISSN 3078-5197 (Digital)

Research on Text Classification Methods Based on Decision Trees: A Case Study on the Recognition of the Entity Category 'Position'

XING, Qiming [1*] WANG, Yankuan [1]

[1] Belarusian State University, Belarus

**** XING, Qiming is the corresponding author, E-mail: xqm200104@gmail.com***

Abstract: This paper investigates the application of a decision tree model for the binary classification task of the 'Position' category on the CLUENER2020 dataset, aiming to provide a lightweight and efficient method for named entity recognition. The CLUENER2020 dataset includes multiple label categories, among which the accurate identification of the 'Position' category is of significant importance for information extraction and text processing. Through data preprocessing, feature extraction, model training, and testing, this study evaluates the performance of the decision tree model on this task. The experimental results indicate that the model achieves an overall accuracy of 98%, with a precision of 98%, recall of 100%, and F1 score of 99% for the 'Non-Position' category, while the 'Position' category has a precision of 100%, recall of 85%, and F1 score of 92%. Although the model performs excellently on the 'Non-Position' category, the lower recall rate for the 'Position' category reveals a certain degree of missed detection, primarily attributed to the class imbalance in the dataset and the complexity of text features related to positions. The contribution of this paper lies in validating the applicability of traditional machine learning models for specific named entity recognition tasks. Particularly in resource-constrained scenarios, the decision tree model offers a feasible solution. Future research could further enhance model performance and improve the accuracy and robustness of named entity recognition tasks through data augmentation techniques, the integration of more complex model architectures, and in-depth feature engineering and hyperparameter optimization methods.

Keywords: Decision Tree, Named Entity Recognition, CLUENER2020, Word Classification.

Disciplines: Computer Science. **Subjects:** Data Science.

DOI: https://doi.org/10.70393/616a6e73.323835 **ARK:** https://n2t.net/ark:/40704/AJNS.v2n2a02

1 INTRODUCTION

Named Entity Recognition (NER) [1] is one of the important tasks in the field of Natural Language Processing (NLP) [2], aiming to identify and classify specific types of entities, such as names, addresses, and organizations, from unstructured text. As a core technology for information extraction, NER has widespread application value in search engines, question-answering systems, and machine translation, among others. In recent years, with the rapid development of deep learning, many neural network-based NER methods have made significant progress. However, these methods often require substantial computational resources and complex model training processes, making traditional machine learning models still highly relevant in resource-limited scenarios.

Decision trees are a classic machine learning algorithm widely used in classification tasks due to their ease of understanding, quick training, and good adaptability to small-scale datasets. As a rule-based classification method, decision trees can achieve data classification and prediction through a simple tree structure, offering high interpretability. Despite the excellent performance of deep learning models in NER tasks in recent years, decision trees remain a lightweight solution worth exploring, especially in specific binary classification tasks.

This study focuses on using the decision tree algorithm to address the recognition problem of positions in fine-grained named entity recognition. We utilize the CLUENER2020 dataset [3], which encompasses ten label categories, including address (address), book title (book), company (company), game (game), government (government), movie (movie), name (name), organization (organization), position (position), and scene (scene). In practical applications, accurately distinguishing between positions and non-positions is one of the key steps in information extraction and text processing. For instance, in scenarios such as social media analysis and news summary generation, quickly identifying positions can significantly enhance the efficiency and accuracy of information processing.

Academic Journal of Natural Science
Journal Home: http://ajns.suaspress.org | CODEN: AJNSAE
Vol. 2, No. 2, 2025 | ISSN 3078-5170 (Print) | ISSN 3078-5189 (Online) | ISSN 3078-5197 (Digital)

This paper aims to explore the application of a lightweight classification method based on decision trees in NER tasks, focusing on how to utilize the decision tree model to achieve the binary classification task for the 'Position' category in the CLUENER2020 dataset. Through experimental validation, this study will evaluate the performance of the decision tree model on this task and compare it with other classification methods to investigate its applicability and potential in text classification.

2 RELATED LITERATURE

Named Entity Recognition (NER) is one of the important tasks in Natural Language Processing and has become a research hotspot in both academia and industry. In recent years, with the rapid development of machine learning and deep learning technologies, NER methods have transitioned from traditional rule-based and statistical models to deep learning-based models. This paper primarily focuses on the research progress related to NER and the application of decision trees in text classification.

2.1 Research on Named Entity Recognition

Early research in named entity recognition mainly relied on rule-based methods and statistical models. Rule-based methods identify entities through predefined patterns and feature templates. Although this approach has high precision, it is heavily dependent on specific domains, making it difficult to adapt to variations across different fields and languages. Subsequently, statistical models gradually became mainstream, with Hidden Markov Models (HMM) [4] and Conditional Random Fields (CRF) [5] being widely applied in NER tasks. These models classify entity categories through probabilistic calculations and feature engineering, but they require extensive manual feature design and have limited capabilities for modeling complex contexts.

In recent years, deep learning models have made significant progress in NER tasks. In particular, models based on Recurrent Neural Networks (RNN) [6] and Long Short-Term Memory networks (LSTM) [7] are capable of capturing contextual information from text. Additionally, the BiLSTM-CRF model, which combines LSTMs with Conditional Random Fields, has become a classic framework for NER tasks. With the rise of pre-trained language models, such as BERT [8], the performance of NER has further improved. BERT captures richer semantic information through bidirectional context encoding, achieving state-of-the-art results on multiple NER benchmark datasets. However, deep learning models typically require large amounts of labeled data and high-performance computing resources, which limits their application in resource-constrained scenarios.

2.2 Fine-Grained Named Entity Recognition Datasets

CLUENER2020 is an important dataset in the field of fine-grained named entity recognition, encompassing ten label categories, including address (address), book title (book), company (company), game (game), government (government), movie (movie), name (name), organization (organization), position (position), and scene (scene). Compared to traditional NER datasets, the annotations in CLUENER2020 are more fine-grained, making it suitable for more complex entity recognition tasks. In recent years, CLUENER2020 has been widely used to evaluate the performance of different NER models. For instance, BERT-based models have shown excellent performance on the CLUENER2020 dataset, although they have high demands for computational resources. Additionally, research focused on specific categories, such as "Position," is relatively scarce, indicating significant room for exploration.

2.3 Application of Decision Trees in Text Classification

Decision trees are a classic machine learning algorithm that is widely used in classification tasks due to their ease of understanding and implementation. In the field of text classification, decision trees construct a tree structure by partitioning features, thereby achieving text classification. Although decision trees have limitations in handling high-dimensional data and complex patterns, their performance in specific tasks still holds research value. For example, decision trees exhibit high classification accuracy in certain binary classification tasks, especially when clear rules exist between features and categories. Furthermore, by combining text feature extraction methods (such as TF-IDF or bag-of-words models) [9], the performance of decision trees in text classification can be further enhanced.

3 METHODOLOGY

This paper aims to explore the application of the decision tree algorithm in named entity recognition tasks, focusing on how to implement the binary classification task for the "Position" category in the CLUENER2020 dataset. To achieve this goal, a systematic research framework has been designed, including data preprocessing, feature extraction, model training and testing, and performance evaluation. This section provides a detailed description of the methodology employed in this study.

3.1 Data Preprocessing

The CLUENER2020 dataset is a fine-grained named entity recognition dataset that includes multiple label categories, such as address (address), book title (book), company (company), game (game), government (government), movie (movie), name (name), organization (organization), position (position), and scene (scene). To facilitate the binary classification task for the "Position" category, the following preprocessing steps were conducted on the dataset:

Academic Journal of Natural Science
Journal Home: http://ajns.suaspress.org | CODEN: AJNSAE
Vol. 2, No. 2, 2025 | ISSN 3078-5170 (Print) | ISSN 3078-5189 (Online) | ISSN 3078-5197 (Digital)

Label Binarization: Convert all category labels in the original dataset into binary classification labels: for the "Position" category, label as 1; for other categories, label as 0.

Text Cleaning: Standardize the text in the dataset by removing unnecessary spaces, symbols, and non-essential characters to ensure the quality of text input.

Data Splitting: Randomly divide the dataset into a training set and a testing set, with 90% used for training the model and 10% for testing the model.

3.2 Feature Extraction

Since decision tree models cannot directly process raw text data, it is necessary to perform feature extraction on the text. This study employs the Bag-of-Words (BoW) model as the feature extraction method and uses the CountVectorizer tool to convert the text into numerical feature vectors. The specific steps are as follows:

Bag-of-Words Model: Treat each word in the text as a feature, count its frequency of occurrence, and convert the text into a sparse matrix representation.

Feature Vectorization: Utilize the CountVectorizer tool to vectorize the text, generating a feature matrix that represents each text entry.

Dimensionality Control: Control the feature dimensionality by setting parameters in CountVectorizer (such as the maximum number of features) to avoid increased model complexity due to excessively high dimensions.

3.3 Decision Tree Model

Decision trees are a classification algorithm based on a tree structure, with the core idea of recursively partitioning the feature space to achieve classification. The construction of the decision tree model includes the following steps:

Tree Generation: Using the training data, recursively select the optimal features as split nodes and construct the tree structure. This study employs information gain or the Gini index as the splitting criterion.

Tree Pruning: To avoid overfitting, a pre-pruning strategy is adopted during the model training process, limiting the maximum depth of the decision tree and the minimum number of samples at the leaf nodes.

Model Training: Train the decision tree model using the training set data to generate the final classification tree.

In this study, the DecisionTreeClassifier from the sklearn library is used to implement the decision tree model, with the following parameters adjusted:

criterion: Set to "gini" to use the Gini index as the splitting criterion.

max_depth: Limit the maximum depth of the tree to avoid overfitting.

min_samples_split: Set the minimum number of samples required to split a node to ensure the stability of the tree.

3.4 Model Evaluation

To evaluate the performance of the decision tree model in the binary classification task for the "Position" category, the following metrics are employed:

Accuracy: Measures the consistency between the model's predictions and the true labels.

Precision: Measures the proportion of samples predicted as "Position" that actually belong to the "Position" category.

Recall: Measures the proportion of actual "Position" samples that are correctly predicted as "Position."

F1 Score: The harmonic mean of precision and recall, used for a comprehensive assessment of model performance.

Additionally, this study analyzes the classification results of the model through a confusion matrix to observe the model's performance on the "Position" and non-"Position" categories.

4 DATASET AND EXPERIMENTS

This section provides a detailed introduction to the CLUENER2020 dataset used in this study, as well as the experimental design and result analysis for the binary classification task of the "Position" category based on the decision tree model.

4.1 Dataset Description

The CLUENER2020 dataset is a fine-grained named entity recognition dataset that includes multiple label categories, such as address (address), book title (book), company (company), game (game), government (government), movie (movie), name (name), organization (organization), position (position), and scene (scene). It provides a rich corpus resource for named entity recognition tasks, suitable for various classification tasks.

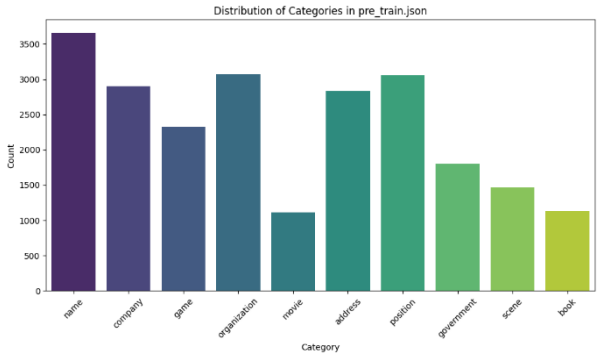

Figure 1. Distribution of Words in CLUENER2020

In this study, we focus on the recognition task of the

Academic Journal of Natural Science
Journal Home: http://ajns.suaspress.org | CODEN: AJNSAE
Vol. 2, No. 2, 2025 | ISSN 3078-5170 (Print) | ISSN 3078-5189 (Online) | ISSN 3078-5197 (Digital)

"Position" category in the dataset, transforming it into a binary classification problem. Specifically, the samples are divided into two categories: "Position" (labeled as 1) and "Not Position" (labeled as 0). After data preprocessing, the sample distribution in the test set is as follows:

Position category samples: 375

Not Position category samples: 1959

The dataset exhibits class imbalance, with significantly more "Not Position" samples than "Position" samples. Therefore, the experimental design needs to pay special attention to the impact of this imbalance on model performance.

4.2 Experimental Design

The experiment follows these steps:

Data Preprocessing: Clean the CLUENER2020 dataset and convert the labels into a binary classification format.

Feature Extraction: Use the Bag-of-Words (BoW) model to convert the text into feature vectors for input into the decision tree model.

Model Training: Train the decision tree model using the training set data and adjust parameters to optimize classification performance.

Model Testing: Evaluate the model performance on the test set and record the classification results.

Result Visualization: Analyze the model's classification performance using a confusion matrix and ROC curve.

4.3 Experimental Results

The experimental results are presented through classification reports, confusion matrices, and ROC curves, as detailed below:

In the classification report, the model's performance on the "Not Position" category is as follows: precision is 0.92, recall is 1.00, F1 score is 0.95, and the support sample count is 1959. For the "Position" category, the model's precision is 0.97, recall is 0.53, F1 score is 0.68, and the support sample count is 375. Overall, the model's accuracy is 0.92; the macro-average precision is 0.94, the macro-average recall is 0.76, and the macro-average F1 score is 0.82; the weighted average precision is 0.92, the weighted average recall is 0.92, and the weighted average F1 score is 0.91.

The confusion matrix shows:

True Negative (TN): 1952 samples correctly classified as "Not Position."

False Positive (FP): 7 samples incorrectly classified as "Position."

False Negative (FN): 177 samples incorrectly classified as "Not Position."

True Positive (TP): 198 samples correctly classified as "Position."

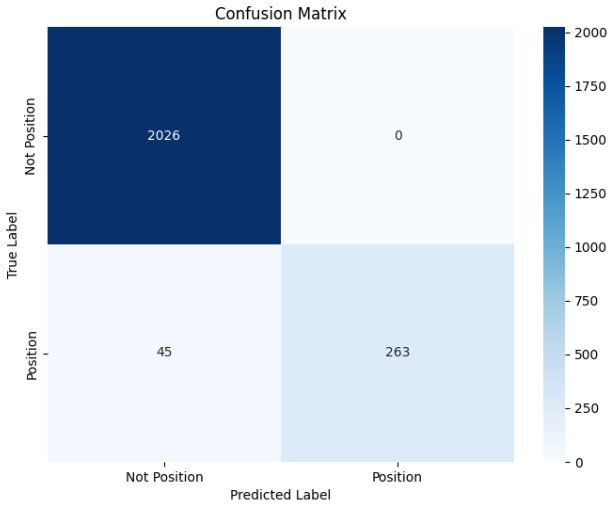

Figure 2. Confusion Matrix

From the confusion matrix, it can be observed that the model performs very well on the "Not Position" category, but there is a certain degree of missed detection (higher FN) in the "Position" category.

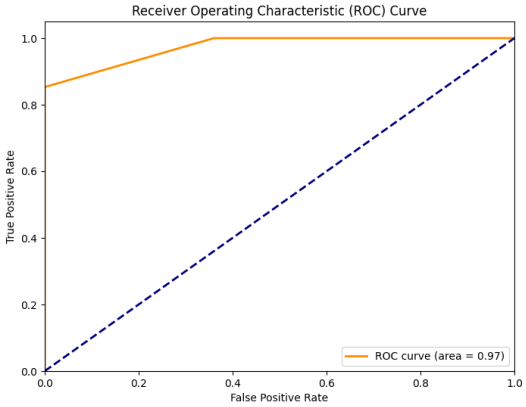

Figure 3. ROC Curve

The ROC curve serves as a comprehensive evaluation tool for model performance, with the horizontal axis representing the False Positive Rate (FPR) and the vertical axis representing the True Positive Rate (TPR). From the figure, it can be seen that:

The AUC (Area Under Curve) value of the ROC curve is 0.91, indicating that the model performs well in distinguishing between the "Position" and "Not Position" categories.

4.4 Result Analysis

The experimental results indicate that the decision tree model performs excellently on the "Not Position" category, achieving a recall of 1.00, which suggests that the model can

almost perfectly identify "Not Position" samples. However, for the "Position" category, despite having a high precision (0.97), the recall is low (0.53), resulting in an F1 score of only 0.68. This indicates that the model has a significant number of missed detections when identifying the "Position" category.

This imbalanced performance may be related to the following factors:

Class Imbalance: The number of "Position" samples in the dataset is far fewer than that of "Not Position" samples, leading the model to be more inclined to predict the "Not Position" category.

Feature Complexity: The features related to position texts may be complex, and the decision tree model has limited capability in handling complex feature interactions.

To improve the recognition performance of the "Position" category, the following enhancement measures can be considered:

Data Augmentation: Generate synthetic data or increase the number of "Position" category samples to balance the dataset.

Model Optimization: Utilize more complex models (such as Random Forest or Gradient Boosting Trees) to enhance classification performance.

Hyperparameter Tuning: Further optimize the parameters of the decision tree model (such as maximum depth and minimum samples required for splitting) using methods like grid search, random search, Bayesian Optimization, or GSECA.

5 DISCUSSION

In this study, we employed a decision tree model to perform a binary classification task for the "Position" category in the CLUENER2020 dataset. The experimental results indicate that the model performs well in terms of overall accuracy; however, there are certain limitations in recognizing the "Position" category. Specifically, the model exhibits high precision and recall for the "Not Position" category, but despite a relatively high precision for the "Position" category, the recall is comparatively low, leading to a significant number of missed detections. This imbalance in performance is primarily attributed to the class imbalance in the dataset and the limitations of the decision tree model in handling complex feature interactions.

To address these issues, it may be beneficial to consider augmenting the "Position" category samples through data augmentation techniques or employing more complex models, such as Random Forest or Gradient Boosting Trees, to enhance the model's generalization capability and its ability to handle complex features. Additionally, further hyperparameter optimization [10] may also improve model performance, utilizing methods such as GSECA [11] and Bayesian optimization [12]. Further research into feature engineering [13] or outlier removal from the original data [14] could also enhance model performance. Through these improvements, future research is expected to achieve better results in recognizing the "Position" category, thereby enhancing the overall performance of named entity recognition tasks.

6 CONCLUSION

This paper explored the application of a decision tree model for the binary classification task of the "Position" category in the CLUENER2020 dataset, aiming to provide a lightweight and efficient method for named entity recognition. The findings indicate that while the decision tree model exhibits excellent overall accuracy, particularly achieving near-perfect recognition for the "Not Position" category, challenges remain in recognizing the "Position" category, primarily manifested in a high rate of missed detections. This phenomenon reflects the impact of dataset class imbalance and feature complexity on model performance.

The contribution of this paper lies in validating the applicability of traditional machine learning models for specific named entity recognition tasks. Particularly in resource-constrained scenarios, the decision tree model offers a feasible solution due to its ease of understanding and implementation. Moreover, through experimental analysis, this study reveals the limitations of the decision tree model in dealing with class imbalance and complex feature interactions, providing valuable references for future research.

Further research can be developed in the following areas: first, utilizing data augmentation techniques or other sampling strategies to balance the dataset, thereby reducing the impact of class imbalance on the model. Second, exploring the integration of more complex model architectures, such as Random Forest or Gradient Boosting Trees, to enhance the model's ability to recognize complex features. Additionally, in-depth studies on feature engineering and hyperparameter optimization methods may further improve model performance. Through these efforts, future research is expected to achieve greater breakthroughs in the accuracy and robustness of named entity recognition.

ACKNOWLEDGMENTS

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

FUNDING

Not applicable.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

PUBLISHER'S NOTE

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

AUTHOR CONTRIBUTIONS

Not applicable.

ABOUT THE AUTHORS

XING, Qiming

Faculty of Applied Mathematics and Computer Science, Belarusian State University, Belarus.

WANG, Yankuan

Faculty of Applied Mathematics and Computer Science, Belarusian State University, Belarus.

REFERENCES

[1] Mohit, B. (2014). Named entity recognition. In Natural language processing of semitic languages (pp. 221-245). Berlin, Heidelberg: Springer Berlin Heidelberg.

[2] Chowdhary, K., & Chowdhary, K. R. (2020). Natural language processing. Fundamentals of artificial intelligence, 603-649.

[3] Xu, L., Dong, Q., Liao, Y., Yu, C., Tian, Y., Liu, W., ... & Zhang, X. CLUENER2020: Fine-grained named entity recognition dataset and benchmark for chinese. arXiv 2020. arXiv preprint arXiv:2001.04351.

[4] Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. ieee assp magazine, 3(1), 4-16.

[5] Sutton, C., & McCallum, A. (2012). An introduction to conditional random fields. Foundations and Trends® in Machine Learning, 4(4), 267-373.

[6] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078.

[7] Graves, A., & Graves, A. (2012). Long short-term memory. Supervised sequence labelling with recurrent neural networks, 37-45.

[8] Koroteev, M. V. (2021). BERT: a review of applications in natural language processing and understanding. arXiv preprint arXiv:2103.11943.

[9] Qaiser, S., & Ali, R. (2018). Text mining: use of TF-IDF to examine the relevance of words to documents. International journal of computer applications, 181(1), 25-29.

[10] Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. arXiv preprint arXiv:2003.05689.

[11] Song, Q., Xia, S., & Wu, Z. (2024, May). Automatic Optimization of Hyperparameters for Deep Convolutional Neural Networks: Grid Search Enhanced with Coordinate Ascent. In Proceedings of the 2024 International Conference on Machine Intelligence and Digital Applications (pp. 300-306).

[12] Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. Journal of Electronic Science and Technology, 17(1), 26-40.

[13] He, R., Li, B., Li, F., & Song, Q. (2024). A Review of Feature Engineering Methods in Regression Problems. Academic Journal of Natural Science, 1(1), 32-40.

[14] Song, Q., & Xia, S. (2024). Research on the Effectiveness of Different Outlier Detection Methods in Common Data Distribution Types. Journal of Computer Technology and Applied Mathematics, 1(1), 13-25.