Auto Questions Tagging

Session: 2021 – 2025

Submitted by:

Muhammad Usman Asghar 2021-CS-46

Submitted to:

Prof. Dr. Usman Ghani

Department of Computer Science

University of Engineering and Technology

Lahore

Table of Contents

Introduction

Online forums and Q&A platforms contain a vast amount of knowledge in the form of questions and answers. To organize and retrieve this information efficiently, tagging systems are employed. However, assigning tags manually can be time-consuming and inconsistent. This project aims to automate the tagging process by classifying questions into predefined tags based on their textual content and performing sentiment analysis to understand the emotional tone of questions and answers.

Research Gaps

Despite advancements in text classification and sentiment analysis, several challenges persist:

1. **Limited Focus on Tag Relevance**: Existing studies often prioritize the overall textual content for classification, ignoring the significance of specific sections (e.g., titles or questions) that might be more relevant to tag prediction.
2. **Lack of Sentiment Analysis Integration**: Sentiment analysis is rarely incorporated into tagging systems, despite its potential to provide insights into user engagement and emotional tone.
3. **Inconsistent Performance Across Models**: Most approaches rely on a single model, which may not perform uniformly across diverse datasets.

This project addresses these gaps by:

- Combining textual content from titles, questions, and answers for comprehensive feature extraction.
- Incorporating sentiment analysis as an auxiliary feature.
- Comparing multiple machine learning models to identify the most effective classifier for tag prediction.

Methodology

Dataset Preparation

Three datasets—**Questions**, **Answers**, and **Tags**—were combined to create a unified dataset:

1. **Questions.csv**: Contains information about question titles, descriptions, and metadata.
2. **Answers.csv**: Aggregated to form a single response per question.
3. **Tags.csv**: Merged to associate questions with their respective tags.

The datasets were preprocessed to remove irrelevant columns, handle missing values, and group tags and answers.

Preprocessing Steps

1. **Text Cleaning**:

    - Removal of punctuation and HTML tags.
    - Conversion to lowercase for consistency.

2. **Stopword Removal**: Used NLTK's stopwords list to eliminate common, non-informative words.
3. **Lemmatization**: Transformed words into their base forms using WordNetLemmatizer.
4. **Sentiment Analysis**: Calculated sentiment polarity scores for titles, questions, and answers using TextBlob.

Feature Extraction

1. **TF-IDF Vectorization**:
    - Applied on titles, questions, and answers to represent text data numerically.
2. **Sentiment Polarity**:
    - Incorporated as additional features for each record.

Model Selection

Six machine learning models were evaluated:

1. **K-Nearest Neighbors (KNN)**
2. **Support Vector Machine (SVM)**
3. **Random Forest Classifier**
4. **Decision Tree Classifier**
5. **Gradient Boosting Machine (GBM)**
6. **Logistic Regression**

Evaluation Metrics

- **Accuracy**: Primary metric for evaluating classification models.
- **Precision, Recall, F1-Score**: Additional metrics for analyzing model performance.

Implementation Details

1. **Data Preprocessing**:

    - Each textual field was cleaned, tokenized, lemmatized, and stripped of stopwords.
    - Sentiment scores were calculated as an auxiliary feature, providing additional contextual information.

2. **TF-IDF Transformation**:

    - Vectorized all text fields to capture term importance and frequency.

3. **Dataset Splitting**:

    - Data was divided into training and testing sets with a ratio of 60:40 for model evaluation.

4. **Hyperparameter Tuning**:

    - Grid search and cross-validation techniques were applied for model optimization.

Results

Sentiment Analysis

Sentiment analysis revealed distinct patterns in the emotional tone of textual data:

- Titles and questions had a generally neutral or positive tone.
- Answers exhibited a slightly higher variance in sentiment polarity, reflecting diverse user perspectives.

KNN Analysis

The accuracy of KNN fluctuated with the number of neighbors (k). The optimal performance was observed at .

Gradient Boosting

Gradient Boosting demonstrated the highest accuracy of 82.7%. Its ensemble nature allowed it to capture complex patterns in the data.

Comparison with Existing Work

Feature	Existing Work	This Project
Tag Prediction Focus	Limited to titles	Integrated multiple features (title, question, answer)
Sentiment Analysis	Rarely included	Incorporated as auxiliary feature
Model Evaluation	Limited to single model	Comparative analysis across six models

Our results demonstrated a significant improvement in accuracy and robustness, with Gradient Boosting outperforming other models. Incorporating sentiment analysis added a nuanced understanding of user interactions, which was not present in prior studies.

Visualizations

1. **KNN Accuracy vs. K Value**:
    - A graph showed accuracy trends as the number of neighbors increased, peaking at .
2. **Model Comparison**:
    - Bar plots illustrated the performance of each model, highlighting Gradient Boosting's superior accuracy.

Feature Importance

Using feature importance analysis, TF-IDF vectorized features were identified as the most significant contributors to the model's performance. Sentiment scores provided additional, albeit less significant, insights.

Discussion

Key Findings

1. **Importance of Feature Integration**:
    - Combining titles, questions, and answers improved classification accuracy.
2. **Sentiment Analysis Insights**:
    - Sentiment polarity scores provided valuable context for understanding user-generated content.
3. **Model Selection**:
    - Gradient Boosting emerged as the best-performing model, balancing accuracy and computational efficiency.

Challenges

1. **Imbalanced Data**:
    - Certain tags were underrepresented, requiring minimum tag occurrence thresholds.
2. **High Dimensionality**:
    - TF-IDF vectorization produced large feature sets, necessitating efficient algorithms.

Limitations

- The dataset was filtered to questions with high scores (>7) and frequently occurring tags (≥1100). This might have excluded niche content.
- The project relied on static thresholds for preprocessing and tag selection.

Recommendations

1. **Improved Dataset Diversity**:
    - Future studies should include questions of varying scores and tags with lower frequencies.
2. **Advanced Modeling Techniques**:

- Consider employing transformer-based architectures (e.g., BERT) for improved text understanding.

3. **Real-Time Implementation**:

    - Build a real-time tagging system for live applications on Q&A platforms.

Conclusion

This project presented a comprehensive approach to classifying questions into tags and analyzing their sentiment. By integrating multiple textual features, applying robust preprocessing techniques, and evaluating diverse machine learning models, we achieved significant improvements in tag prediction accuracy.

Future Work

1. **Expand Dataset Scope**:

    - Include questions with lower scores and less frequent tags to enhance model generalizability.

2. **Deep Learning Models**:

    - Explore transformer-based architectures like BERT for improved semantic understanding.

3. **Real-Time Implementation**:

    - Develop a real-time tagging system for live Q&A platforms.

References

1. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
2. TextBlob Documentation. (n.d.). Retrieved from https://textblob.readthedocs.io/
3. Scikit-learn Documentation. (n.d.). Retrieved from https://scikit-learn.org/
4. Vaswani, A., et al. (2017). *Attention Is All You Need*. Advances in Neural Information Processing Systems.