# NLP701 / 805 Assignment 2 - Multilingual Characterization and Extraction of Narratives from Online News

**Sarfraz Ahmad (24010739)**
SemEval 2025 Task ID: Sarfraz307
Natural Language Processing Department,
Mohamed bin Zayed University of Artificial Intelligence,
Masdar City, Abu Dhabi
sarfraz.ahmad@mbzuai.ac.ae

## Abstract

Entity framing in news articles shapes public perception by assigning individuals, groups, or organizations roles such as protagonists, antagonists, or innocents. SemEval-2025 Subtask-A focuses on a multi-label, multi-class text-span classification task to assign fine-grained roles to Named Entities (NEs) in news articles. Using the article text, entity mentions, and their span offsets, the task involves labeling each entity based on a taxonomy encompassing protagonists, antagonists, and innocents, including sub-roles. Challenges include imbalanced role distributions, contextual role dependencies, and the complexity of multi-label classification. This work explores robust methods leveraging contextual cues and entity-specific information to enhance understanding of narrative structures and media framing. [1]

## 1 Introduction

Entity framing in news articles significantly influences public perceptions by portraying individuals, groups, or organizations in specific roles, such as protagonists, antagonists, or innocents. These roles align with overarching narratives, reflecting societal contexts and shaping public discourse. By influencing reader sentiment, framing plays a critical role in constructing broader narratives in media.

SemEval-2025 Subtask-A introduces the task of multi-label, multi-class classification of named entities (NEs) in news articles. This involves identifying entity mentions, mapping their span offsets, and assigning one or more roles to each entity based on a predefined taxonomy. The taxonomy categorizes roles into three main types—protagonists, antagonists, and innocents—with fine-grained distinctions for nuanced framing.

The task presents challenges due to imbalanced role distributions, context-dependent role assignments, and the multi-label nature of classification, where entities can simultaneously occupy multiple roles. This work addresses these challenges by leveraging contextual and entity-specific information. Advancements in accurate role classification have applications in media analysis, computational journalism, and narrative modeling, enhancing our understanding of how narratives influence public opinion and discourse.

## 2 Methods

### 2.1 Dataset

SemEval-2025 Subtask 1 focuses on the entity framing task, where training and development (dev) datasets are provided. However, the gold labels for the dev data are not accessible. The training dataset consists of 686 samples with columns including `article_id`, `entity_mention`, `start_offset`, `end_offset`, `main_role`, and `fine_grained_roles`. These samples are derived from 202 unique documents.

The task involves three main roles—`Antagonist`, `Protagonist`, and `Innocent`, which represent a multi-class label. Additionally, the `fine_grained_roles` column contains 22 unique labels and is multi-label in nature. The class distribution for the `main_role` column is as follows: `Antagonist` has 477 samples, `Protagonist` has 130 samples, and `Innocent` has 79 samples. This highlights the data imbalance in the `main_role` column.

To address this, stratified sampling was applied to create a validation set consisting of 10% of the data. The sample distribution across the datasets is as follows: the training set contains 617 samples, the validation set contains 69 samples, and the development set contains 91 samples.

### 2.2 Data Preprocessing

Preprocessing ensures that the datasets are clean, standardized, and ready for modeling. The preprocessing was tailored to capture relevant information

[1] The code is available at git repo: https://github.com/SarfrazAhmad307/Assignment2-NLP701

while ignoring redundancies. After reading the annotations file for the training and development sets, the `article_id` field was used to fetch the corresponding article and populate a new column named `document`. In this new column, entity mentions were identified based on the offsets provided in the annotations. The entity mentions were then used to extract a passage containing the mention as a "span text." This span was selected by applying a delimiter of `\n\n` on both sides of the mention. This method of selection was motivated by the need to capture the context surrounding the entity mention, as it provides a natural segment of text that is more likely to be relevant to the task, rather than arbitrarily truncating the text at mention boundaries. The extracted span was then stored in a new column.

The text in this column was then subjected to several preprocessing steps. First, contractions and punctuation were mapped, and common misspellings were corrected. Emojis and noise such as links, punctuation, text within square brackets, and words containing numbers were removed. The text was then converted to lowercase for proper standardization. Subsequently, stopword removal was applied to the cleaned text, eliminating frequently occurring English words that do not contribute to the semantic meaning of the content. The whole pipeline of task is given in Figure 1 of Appendix A.

### 2.3 Feature Engineering

For this task, four feature engineering approaches were selected to train multiple models and evaluate their performance. The selected techniques include TF-IDF, BERT embeddings, RoBERTa embeddings, and T5 embeddings. Each approach provides unique advantages in representing textual data for classification tasks.

The motivation behind using TF-IDF stems from its effectiveness in capturing term importance within a document while accounting for its frequency across the corpus. This makes it a simple yet robust baseline for text representation. On the other hand, embeddings generated by large language models (LLMs) such as BERT, RoBERTa, and T5 leverage contextualized representations, capturing semantic and syntactic nuances in the text. These embeddings are particularly well-suited for handling the complexities of the current task, including multi-label and multi-class classification, as they provide rich, contextual information for each token and the overall document.

For each of the LLM-based models (BERT, RoBERTa, and T5), embeddings were extracted from the last hidden state. The resulting dimensions of the embeddings were (sample, max sequence length, embedding dimension). This structure captures the sequence of token-level embeddings across the entire document. To make these embeddings compatible with machine learning models, average pooling was applied across the sequence length dimension. This reduced the shape to (sample, embedding dimension), resulting in a fixed-size vector that effectively summarizes the entire document. Hidden state embeddings were used because they capture the deep, contextualized understanding of the text, allowing the model to leverage both syntax and semantics. Average pooling was chosen as it efficiently combines information from all tokens, providing a compact representation while retaining the critical features needed for classification. Equations for both TF-IDF and average pooling are given in Appendix A.

By employing these diverse feature engineering techniques, the goal was to analyze their impact on model performance and determine the most effective representation for this task.

### 2.4 Predictive Models

To predict the `fine_grained_roles` label in a multi-label classification setting, various traditional ML models and LLMs were employed. These models were selected to address the complexities of overlapping roles and imbalanced data distributions.

The ML models used include KNN, Decision Tree, Bagging, Random Forest, Boosting, MNB, Support Vector Classifier (SVC) with squared hinge loss, and the Label Power Set approach with LinearSVC. Each model adopts a unique approach, ranging from instance-based learning (k-NN) and probabilistic modeling (NB) to tree-based methods (Decision Tree, Bagging, Random Forest) and hyperplane optimization (SVC). Boosting improves performance through sequential training of weak learners, while Label Power Set transforms multi-label problems into multiple binary classification tasks.

These models were trained and evaluated to identify the most effective technique (embedding+model) for predicting fine-grained roles, pro-

viding insights into their suitability for this task.

## 3 Results and Evaluation

The performance of machine learning models was evaluated using the Exact Match Ratio (EMR) score for predicting the `fine_grained_roles` label, with results based on the validation set. The outcomes for four embeddings—TF-IDF, BERT, RoBERTa, and T5—are summarized in Table 1, showcasing the impact of embedding choice on model performance.

Across all embeddings, the Label Power Set approach with LinearSVC consistently outperformed other models, achieving the highest EMR scores: 0.3623 with BERT, 0.3478 with both TF-IDF and T5, and 0.3188 with RoBERTa. Contextual embeddings like BERT and T5 generally provided better results than TF-IDF, with BERT demonstrating the strongest overall performance on the validation set. Among traditional models, Decision Tree and ensemble methods (Random Forest and Boosting) performed moderately, while k-NN and Multinomial Naive Bayes underperformed across all embeddings.

After selecting the best combination of embedding and model (BERT + Label Power Set with LinearSVC), we tested our model on the development set. The leaderboard results are given in Table 2

These results highlight the significant influence of embedding choice on model performance, with contextual embeddings yielding superior representations for the task. The Label Power Set approach with LinearSVC proved to be the most effective model for multi-label classification of `fine_grained_roles`.

# Appendix

## A Images

## A Equations

The following equations describe the feature engineering techniques used in this task:

### A.1 TF-IDF Equation

The Term Frequency-Inverse Document Frequency (TF-IDF) is calculated as the product of two components: term frequency (TF) and inverse document frequency (IDF).

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \quad (1)$$

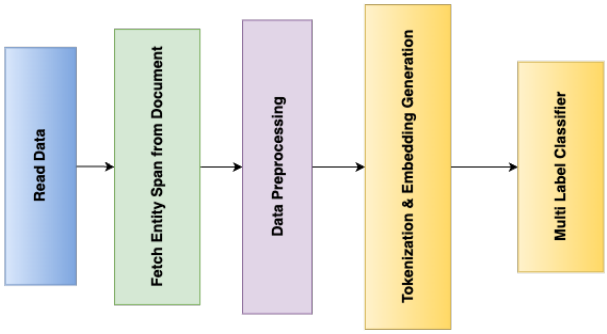


Figure 1: Task Pipeline

Where:

$$\text{TF}(t, d) = \frac{\text{Count of term } t \text{ in document } d}{\text{Total number of terms in document } d}$$

$$\text{IDF}(t) = \log\left(\frac{N}{\text{DF}(t)}\right)$$

Here, $N$ is the total number of documents, and $\text{DF}(t)$ is the number of documents containing term $t$.

### A.2 Average Pooling Equation

For large language models, the last hidden state embeddings of a document are denoted as $H$, where $H \in R^{(S,L,D)}$. Here, $S$ is the number of samples, $L$ is the maximum sequence length, and $D$ is the embedding dimension. Average pooling is applied to reduce the sequence length dimension, resulting in an embedding with shape $R^{(S,D)}$.

The average pooling operation is defined as:

$$\mathbf{h}_{\text{avg}} = \frac{1}{L} \sum_{l=1}^{L} H_{s,l,d} \quad (2)$$

Where:

$$\mathbf{h}_{\text{avg}} \in R^{(S,D)}$$

is the pooled embedding for sample $s$, and $H_{s,l,d}$ is the embedding of the $l$-th token in the $s$-th sample and $d$-th dimension.

The result of this operation is a fixed-length vector for each sample that aggregates the information from all tokens in the document.

## B Validation Set Results

Table 1 presents the Exact Match Ratio (EMR) scores for all models evaluated across the four embeddings on validation set: TF-IDF, BERT, RoBERTa, and T5. These results provide a detailed comparison of the performance of machine learning models in predicting the `fine_grained_roles` label.

Table 1: Exact Match Ratio scores for different models across embeddings

| Embedding | Model | EMR |
|---|---|---|
| TF-IDF | KNN | 0.1159 |
| | Decision Tree | 0.2319 |
| | Bagging | 0.1739 |
| | Random Forest | 0.2029 |
| | Boosting | 0.2029 |
| | MNB | 0.0145 |
| | SVM | 0.2174 |
| | LPS (LinearSVC) | 0.3478 |
| BERT | KNN | 0.0580 |
| | Decision Tree | 0.2464 |
| | Bagging | 0.1884 |
| | Random Forest | 0.2029 |
| | Boosting | 0.2029 |
| | MNB | 0.0145 |
| | SVM | 0.1304 |
| | LPS (LinearSVC) | 0.3623 |
| RoBERTa | KNN | 0.0580 |
| | Decision Tree | 0.2319 |
| | Bagging | 0.1594 |
| | Random Forest | 0.2029 |
| | Boosting | 0.2029 |
| | MNB | 0.0145 |
| | SVM | 0.1014 |
| | LPS (LinearSVC) | 0.3188 |
| T5 | KNN | 0.1014 |
| | Decision Tree | 0.2464 |
| | Bagging | 0.1884 |
| | Random Forest | 0.2029 |
| | Boosting | 0.2029 |
| | MNB | 0.0145 |
| | SVM | 0.0580 |
| | LPS (LinearSVC) | 0.3478 |

## C Appendix: Development Set Results

After selecting the best combination of embedding and model (BERT + Label Power Set with LinearSVC), we tested the model on the development set. The leaderboard results are presented in the table below:

Table 2: Leaderboard results for the best model on the development set.

| Metric | Score |
|---|---|
| Exact Match Ratio | 0.26370 |
| micro P | 0.27470 |
| micro R | 0.25000 |
| micro F1 | 0.26180 |
| Accuracy for Main Role | 0.70330 |

## D Implementation Details

All experiments were conducted using Python with scikit-learn and Hugging Face Transformers libraries. The embeddings (TF-IDF, BERT, RoBERTa, and T5) were generated using pretrained models and fine-tuned where applicable. Stratified sampling was employed to divide the dataset into training and validation sets. Evaluation was based on the Exact Match Ratio metric, which measures the proportion of samples with all labels correctly predicted.