# Magnum JUCSE at SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes

**Adnan Khurshid** and **Dipankar Das**
Department of Computer Science and Engineering
Jadavpur University, Kolkata, 700032, India

## Abstract

This paper explores the detection of persuasion techniques within meme text, emphasizing logical fallacies and emotional appeals. Using a multilingual dataset structured as a directed acyclic graph, the study employs a node-level hierarchical classification with Support Vector Machines and pretrained sentence embeddings. Results demonstrate effective capture of nuanced persuasion techniques, providing fine-grained and general labels. The paper acknowledges dataset imbalance and assesses threshold impact on classification. The work contributes to understanding memes as conduits for persuasive communication, paving the way for future integration of image information for comprehensive analysis.

## 1 Introduction

In the realm of digital communication and social media, memes have emerged as a powerful and widely shared form of content, known for their ability to convey messages in a succinct and often humorous manner. While memes are commonly associated with entertainment, their potential as a tool for persuasive communication, particularly in the context of textual content, has become increasingly evident. This paper focuses on the nuanced task of detecting persuasion techniques within meme text in multiple languages like English, North Macedonian, Arabic and Bulgarian, exploring the ways in which textual elements contribute to the dissemination of persuasive messages.

The main strategy of the system is to train a binary classifier for each node in the hierarchy and predict labels in a top down fashion by seeing the confidence value of the prediction at any node. For each unique label in the hierarchy, a dataset is created from the original dataset which is then used to train the binary classifier for that label.

This task (Dimitrov et al., 2024) helped in understanding the intricacies of Hierarchical classification as well as sentence transformers. Our team participated in subtask 1 and ranked 21 out of 34 in English Language whereas 4 out of 20 in Bulgarian, 3 out of 20 in North Macedonian and 11 out of 17 in Arabic.

### 1.1 Objectives

The main objectives in this task include achieving the accuracy in classification of the internal nodes and minimising the number of classifiers and to look for a global classifier approach which takes the whole hierarchy into account at once. One more challenge due to having multiple levels of classes is handling the problem of inconsistency in predictions at different levels which means that the system may give negative prediction for some class at a level and then gives positive prediction for its children nodes. Since there are multiple output labels, Instances may belong to multiple classes that are not mutually exclusive or have overlapping characteristics due to the hierarchy being in the form of Directed Acyclic Graph. Distinguishing between such classes becomes complex

### 1.2 Contribution

The work done aims to create a model which performs the task of hierarchical multilabel classification of Persuasion techniques in memes with maximum accuracy. The model not only predicts the leaf nodes but also is able to predict corresponding internal nodes if the confidence in prediction is lower than some specified threshold at some node. Thus solving the class parent-child inconsistency problem stated earlier and providing a more robust and comprehensive classification of persuasion techniques in memes, enabling a deeper understanding of the hierarchical structure and allowing for enhanced decision-making based on varying levels of confidence in the predicted labels.

*Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1015–1018
June 20-21, 2024 ©2024 Association for Computational Linguistics

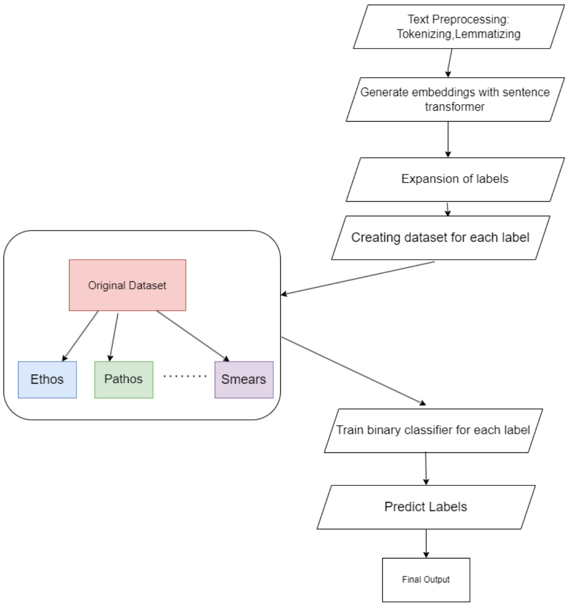
Text Preprocessing: Tokenizing,Lemmatizing
Generate embeddings with sentence transformer
Expansion of labels
Creating dataset for each label
Original Dataset
Ethos
Pathos
Smears
Train binary classifier for each label
Predict Labels
Final Output

Figure 1: A diagram of the workflow.

## 2 Background

Our task involves the detection of persuasion techniques from memes. The data is provided in JSON format with string text and a list of string labels for each text. The labels are defined in a hierarchical manner in the form of a directed acyclic graph (DAG). A training set, a development or validation set, as well as a test set are available. The training and validation sets contain the labels while the test set only contains the text. In our dataset, 29 persuasion labels are defined.

The dataset is quite unbalanced, with some labels occurring many times, with others occurring much less. This can be partially attributed to the hierarchical nature of the data. Labels present neaerer to the root of the label DAG tend to appear much more frequently than the labels present nearer to the leaves of the DAG

## 3 System Overview

The step by step flow of the system is shown in the Figure [1] and explained in detail in the subsections which follow.

### 3.1 Overview

In this work, we use node level hierarchical classification. Our method consists of four major phases, data denoising, feature generation, node level classifier training and finally inference. Initially the data is cleaned and denoised, post this, features are generated for each of the sentences using a pretrained sentence transformer. For classification, we consider a binary classifier at each node (Silla and Freitas, 2011) which predicts whether the example belongs to that node or not. We have employed the SVM (Support Vector Machine) as the classifier in our case.

Inference is done in a top-down fashion which the branch to be taken at each node is decided by the classifier at that node. This allows us to provide fine-grained as well as general labels. Fine grained labels are available toward the leaves of the tree and general labels are available towards the root. Based on the decision probabilities, we select the most suitable depth for the prediction results.

A final point worth mentioning is the identification of the threshold. Due to the imbalanced nature of the dataset , a threshold is determined using trial and error. The system works best with low threshold values for positive class because the training dataset for each unique label becomes highly skewed with negative examples.

### 3.2 Feature generation

Training a large language model from scratch on a corpus of strings requires very heavy computational resources, to which we did not have access. To circumvent this, we have utilized transfer learning, where the embeddings generated form a model on a general task is applied downstream effectively. This allows us to reuse previous work, if the task is sufficiently general, the pretrained model can produce very contextual and high quality embeddings.

For our current work, we have utilized the Sentence Transformer with Siamese BERT Embeddings as described in (Reimers and Gurevych, 2019). The authors of this paper have derived sentence embeddings in a contrastive manner utilizing similarity losses. Namely, they have utilized the triplet loss, which involves the creation of an anchor, a positive pair and a negative pair for embedding generation.

$$\mathcal{L}(A,P,N) = \max\left(d(A,P) - d(A,N) + \alpha, 0\right) \tag{1}$$

The goal of the Triplet Loss function is to minimize the distance between the anchor and the positive sample while simultaneously maximizing the distance between the anchor and the negative sample. A classification loss has also been utilized by the authors. Three labels have been considered, contradiction, neutral and entailment between

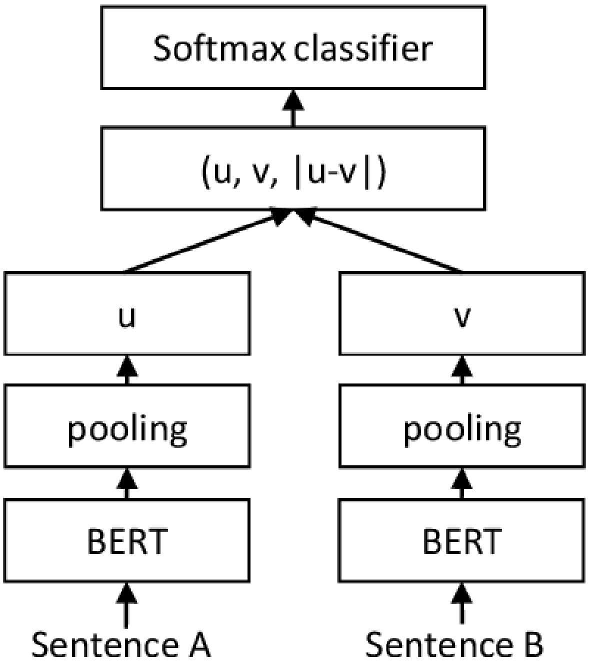

Figure 2: Architecture of the utilized sentence transformer.

pairs of sentences. The generated embeddings have length 768.

where:

- $A$ represents the anchor sample,
- $P$ represents the positive sample (same class as anchor),
- $N$ represents the negative sample (different class from anchor),
- $d(A, P)$ denotes the distance between anchor and positive sample,
- $d(A, N)$ denotes the distance between anchor and negative sample,
- $\alpha$ is the margin, a hyperparameter that specifies the minimum difference between the distances.

### 3.3 Node Level Classifier

The data labels are represented in the form of a DAG. At each node, a SVM(support vector machine) is trained to predict whether the text instance belongs to that node or not. The node level classifiers are trained on the feature embeddings generated using the pretrained sentence transformer.

Support Vector Machines (SVMs) are powerful supervised learning models used for classification and regression tasks. The fundamental idea behind SVMs is to find the hyperplane that best separates the data points into different classes while maximizing the margin, which is the distance between the hyperplane and the nearest data points of each class.

SVMs can handle linearly separable as well as non-linearly separable data by employing the kernel trick, which maps the input data into a higher-dimensional space where it is easier to find a separating hyperplane. The optimization problem associated with SVM can be formulated as a convex optimization problem, typically solved using techniques such as quadratic programming.

## 4 Experimental Setup

### 4.1 Data Preprocessing

The input data is in the form of textual content for memes. There is a mix of capitalized, uncapitalized data as well as non-English words and gibberish. There is also the presence of arbitrary newlines in the dataset. To clean this data, firstly we have removed the unnecessary newlines in the data, replacing them with a single white-space , post this, we have removed all the punctuation. After this, we have lowercased the all the strings in the dataset, followed by stopword removal and lemmatization. This preprocessing improves the performance of the model as in general the dataset is very noisy and a model trained on it will not perform up to the mark.

### 4.2 Dataset Splitting

The original dataset is expanded by adding all the labels from root to leaf for a specific leaf label. So for example, if a row has label 'Slogans', then all the labels from root (Persusaion) to leaf (Slogans) are added, namely, Persuasion, Logos, Justification, Slogans and thus a dataset with expanded labels is formed. The dataset is then represented in One-Hot Encoding format for all the unique labels in the Hierarchy. So the dataset now contains 31 columns, 1 for the text, 1 for embeddings and 29 columns for the 29 labels in the hierarchy. So if a row has labels [Persuasion, Logos, Justification, Slogans] then the columns of these labels will have value 1 and others will have 0. Then a set of smaller datasets with the columns text,embeddings and the binary output for each label is created from the original dataset. These datasets are stored in a dictionary in key-value pairs where the key is the label and value is a dataframe containing the dataset. Thus for the 29 unique labels in the hierarchy, 29 datasets are created.

### 4.3 Inference

The classification is done in two ways. First the text embedding is passed through all leaf node classifiers and the labels which give positive prediction with confidence greater than 0.7 are directly added to the output. Secondly, we then pass the embedding to a function which does the classification in a top down or depth first approach. We start from the root by pushing the children nodes of the node which has positive prediction confidence greater than the predefined threshold value to a stack. Then we pop from the stack and keep repeating until a leaf node is reached or the prediction confidence is very low at any particular node. The distinct labels from both these are then taken as the final output.

## 5 Results

We have provided the results of our method using some different thresholds. The result contains Hierarchical F1 Score, Hierarchical Precision as well as Hierarchical Recall. How the threshold is set is explained in the table [1]. A confusion matrix is shown for the prediction of leaf nodes in Figure [3] The test results for the languages Bulgarian and North Macedonian after final submission are also shown in tables [2] and [3]

| Threshold | Hierarchical F1 | Precision | Recall |
|---|---|---|---|
| For Depth = 0 : 0.3<br>For Depth = 1 : 0.4<br>For Depth $\geq$ 2 : 0.5 | 0.5624 | 0.6322 | 0.5065 |
| All nodes : 0.24 | 0.6034 | 0.5465 | 0.6734 |

Table 1: Results for different threshold values at different depths of the hierarchy.

| Threshold | Hierarchical F1 | Precision | Recall |
|---|---|---|---|
| All nodes : 0.24 | 0.49986 | 0.47027 | 0.53342 |

Table 2: Final submission result on test data in Bulgarian Language.

| Threshold | Hierarchical F1 | Precision | Recall |
|---|---|---|---|
| All nodes : 0.24 | 0.48267 | 0.48568 | 0.47970 |

Table 3: Final submission result on test data in North Macedonian Language.

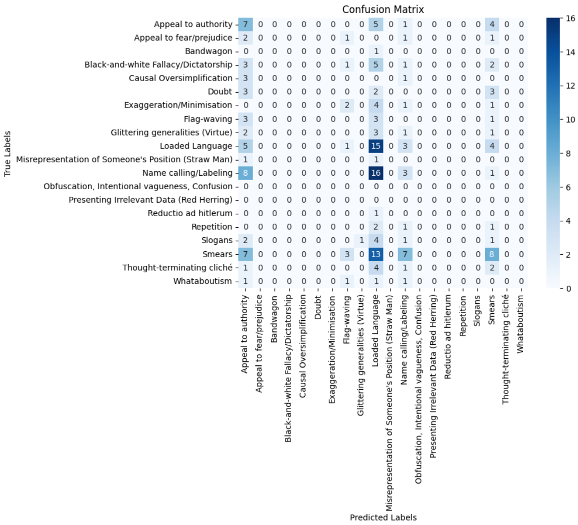

Figure 3: Confusion matrix for only leaf node predictions.

## 6 Conclusion and Future Work

The system gives satisfactory results on the validation dataset but more testing is required to measure the accuracy of the model. The accuracy of classifiers for some of the internal nodes is low because of a large variety of text sentences corresponding to the internal labels The leaf node classifiers generally have very high accuracy due to low number of example instances

This system only works with textual data, considering memes have rich image information as well, utilizing it in sync with the textual data to accurately predict persuasion techniques would be a natural continuation of this work.

## References

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data mining and knowledge discovery*, 22:31–72.