# Fralak at SemEval-2024 Task 4: combining RNN-generated hierarchy paths with simple neural nets for hierarchical multilabel text classification in a multilingual zero-shot setting

**Katarina Laken**
Fondazione Bruno Kessler / Trento, Italy
Universidade de Santiago de Compostela / Santiago de Compostela, Spain
alaken@fbk.eu

## Abstract

This paper describes the submission of team fralak for subtask 1 of task 4 of the Semeval-2024 shared task: 'Multilingual detection of persuasion techniques in memes'. The first subtask included only the textual content of the memes. We restructured the labels into strings that showed the full path through the hierarchy. The system includes an RNN module that is trained to generate these strings. This module was then incorporated in an ensemble model with 2 more models consisting of basic fully connected networks. Although our model did not perform particularly well on the English only setting, we found that it generalized better to other languages in a zero-shot context than most other models. Some additional experiments were performed to explain this. Findings suggest that the RNN generating the restructured labels generalized well across languages, but preprocessing did not seem to play a role. We conclude by giving suggestions for future improvements of our core idea.

## 1 Introduction

Task 4 of the Semeval 2024 workshop deals with the identification of persuasion techniques in meme data (Dimitrov et al., 2024). Subtask 1 regarded only the textual content of the memes. Training, validation and development data is only available in English, but the test phase includes data in three more languages (Bulgarian, North Macedonian and Arabic) for multilingual zero-shot classification. The 20 persuasion techniques are organized in an hierarchical directed acyclic graph (available on the task website). Each meme can have zero, one or multiple persuasion techniques associated with it, making this a hierarchical multilabel classification problem. Assigning a parent node of the target label results in partial points.

Our system (team fralak) implements an ensemble model including a seq2seq module, using some innovations to avoid common pitfalls and exploit the hierarchy information. Our approach transforms the problem by restructuring the labels into strings in a way that captures all possible paths through the hierarchy and uses these as target sequences to train a RNN that learns the relationships between the labels on different levels. It combines the power of a RNN with a simple fully connected architecture. These non-sequential modules are also expected to mitigate the error propagation effect (also called exposure bias), where a wrongly predicted label in the beginning of the generated sequence results in more errors down the line (Xiao et al., 2021). RNNs for multilabel classification also depend on the ordering of the labels, even though the class labels are essentially an unordered set (Wang et al., 2021a). We address this by sorting the labels by frequency. The textual content of the memes is represented using multilingual sentence embeddings. Although we only participated in subtask 1, our architecture can easily be expanded to also take into account the visual content of the meme (subtask 2a+b)[1]. Our system performed below average on the English-only test set, but generalized better than most other systems. The goal of this paper is to explain our methodology and system architecture (sections 2 and 3) and explore why the system performs relatively well at the zero-shot task (section 5).

### 1.1 Background

Hierarchical multilabel classification is applied in many domains, from biology (genomics) (Romero et al., 2023; Wang et al., 2021b) to the classification of images (Lanchantin et al., 2021) or text data (Xiao et al., 2021; Omar et al., 2021). A challenge of this type of data is that the data is virtually always unbalanced on all levels of the hierarchy (Tarekegn et al., 2021). Labels also tend to be correlated.

[1] although the performance of the system on these multimodal tasks remains to be seen



*Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 596–601

There are several approaches to hierarchical multilabel classification. Some studies transform the problem, for example by creating a chain of binary classifiers, whereas others adapt the classification algorithm (Bogatinovski et al., 2022). Some approaches construct a model for each label, but this becomes very computationally expensive: once the amount of labels grows, it is difficult for labels with very few instances, and it is difficult to capture relationships between the labels (Chen and Ren, 2021). The hierarchy can be leveraged for classification. For example, Giunchiglia and Lukasiewicz (2020) use prediction on the lower classes in the hierarchy to make predictions on the upper ones. Seq2seq models are popular for multilabel classification (Chen and Ren, 2021; Chen et al., 2023; Huang et al., 2021). The main idea behind the employment of seq2seq models is that they are able to capture the correlations between labels (Chen and Ren, 2021). Huang et al. (2021) found that a seq2seq model using a biLSTM outperformed other SOTA approaches using chains of classifiers.

The past years have seen the rise of transfer learning, where some model is used for the classification of a different type of data than the data it was trained on (Iman et al., 2023). A common approach to multilingual transfer learning is the use of mapping words or sentences to vectors in a vector space that aligns embeddings for different languages. Training some model on these representations in language A then allows it to make predictions about data in unseen data B, as long as its embeddings are meaningfully mapped to the same vector space (Reimers and Gurevych, 2019).

## 2 Methodology

We aimed to implement rather simple NN modules in order to explore their usefulness for a complicated task like this. The main idea behind our system is to transform the labels into strings that reflect the hierarchical acyclic graph containing the different persuasion techniques. These are be used to train an RNN that is supposed to learn the relationships between both the labels and the different levels of the hierarchy. We expect that the relations between labels are a feature that generalizes especially well across languages, making our approach especially adapt for multilingual zero-shot learning for this specific task.

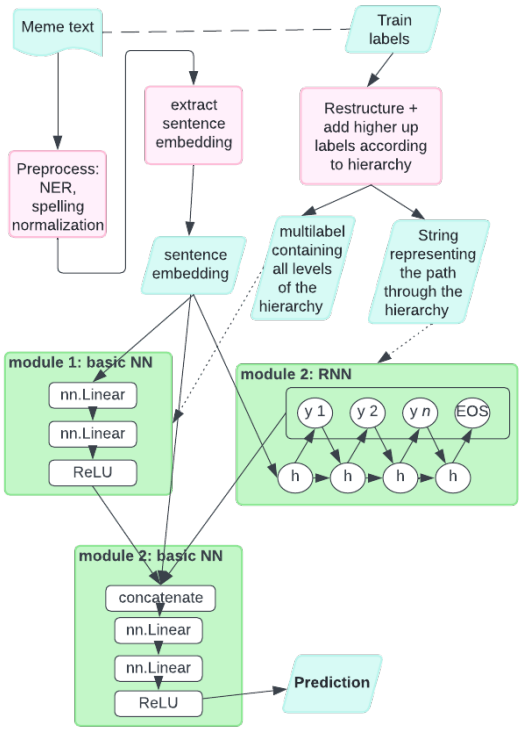


Figure 1: The hierarchical directed acyclic graph containing the persuasion techniques

### 2.1 Preprocessing

The preprocessing consisted of three main steps: spelling normalization, named entity recognition (NER), and adding the sentence embeddings. Since we wanted the system to be able to be applied to other languages as well, preprocessing was limited to some regular expressions capturing characters that repeated more than twice, irregular white spaces, and regularizing different kinds of *haha*'s to a simple 'haha'[2].

The NER was performed using a pretrained multilingual model for token classification by Babel[3], accessed through the Huggingface API. We compiled a list of the 10 people most commonly occurring in the training data and made a dictionary that 'translated' all of their names to one token (so 'Barack Obama' or 'Barack Hussein Obama' etc. would both be 'translated' to 'Obama'). All person entities (recognized with a certainty of over .8) that did not appear in this list were replaced by the name 'Mark', in order for them to be mapped to

[2] t = re.sub(r'[AaHhJjXxAa]*[HhJjXx]?[AaAa]+[HhJjXx] + [AaAa]+[AaHhJjXxAa]*', 'haha', t) ; the double a's are because one is the Cyrillic A and the other is the Latin A

[3] https://huggingface.co/Babelscape/wikineural-multilingual-ner

some kind of baseline name rather than the OOV token[4].

Meme data is expected to contain a lot of non-normative language. We chose to represent it using multilingual sentence embeddings, as these are typically better at dealing with OOV tokens. We used the multilingual variation of the sentence embeddings by (Reimers and Gurevych, 2019)[5].

These embeddings were used as the input for a neural architecture consisting of three separate modules (see 1). Module 1 consisted of a simple neural network (of hidden size 128), trained over 45 epochs, with one input layer, one output layer, and a ReLU activation. The input consisted of the sentence embedding for the meme; the output was a simple multiclass classification with one output for each activated node (meaning the target label and all of its progenitore nodes).

### 2.2 Restructuring of the labels

The central innovation of our approach is the inclusion of hierarchy information by way of transforming the labels to strings reflecting all possible paths through the hierarchy. Module 2 was an RNN that learnt to generate a sequence reflecting the labels and the hierarchy they were embedded in. First, all labels were sorted by frequency in the training data; multiword labels were turned into one-word labels (for example, 'thought terminating cliché' became 'cliché'). We then added the labels from the levels above them (as represented in the label hierarchy graph). As the hierarchy has different levels, this means that every meme was doubled or tripled in the training data, but with different labels. For instance:

**Sequence:** VISIT RUSSIA\n\nBEFORE RUSSIA VISITS YOU
**Label 1:** *Only labels*
'repetition and black and appeal EOS'
**Label 2:** *Labels + red level*
'logos namely repetition and logos namely black and logos and pathos namely appeal EOS'
**Label 3:** *Labels + red level + blue level*
'logos namely repetition and logos namely reasoning namely black and logos namely justification and pathos namely appeal EOS'
**Label 4:** *Labels + red level + blue level + green level*
'logos namely repetition and logos namely reasoning namely simplification namely black and logos namely justification and pathos namely appeal EOS'

The idea of this doubling of labels was that higher-up levels would appear more often and thus become more likely to be predicted by the module. However, preliminary testing showed that it made hardly any difference to use only labels of type 4 or all kinds of labels, likely because lower level labels inherently appear less often due to them governing less nodes.

### 2.3 System architecture

A simple RNN (hidden size = 128) was trained over 25 epochs to generate restructured labels. The model generates labels either until the max string length (manually set to 50) was reached, or until the EOS token was generated[6]. This module was supposed to learn the relationships between labels both at the same and at different levels of the hierarchy; we expected this knowledge to transfer rather well to the unseen multilingual data.

The final module (module 3 in figure 1) concatenates the meme embedding with the outputs of the modules 1 and 2 and the meme embedding and passes it through two fully connected layers (hidden size = 128) and a ReLU activation function (dropout = 0.2). This module, that was trained over 50 epochs, outputs the final prediction of the labels.

## 3 Experimental setup

The training+validation data consisted of 7,500 memes. After restructuring the labels this gave us 21,968 training instances. Including the development data (1,000 memes) resulted in 24,664 instances spread over 8,500 memes. All of these instances were in English. As typical for multilabel settings, the class labels are extremely unbalanced: the most common label, *Smears*, occurs 1,990 times in the training data, whereas the least common label, *Intentional vagueness*, occurred only 21 times. Our teams original test submission was only trained on the train and validation data, as we used the development data to validate our approach, but we added the development data in subsequent experiments as we theorized that having

[4]this step did not take into account different alphabets
[5]Accessed through the Huggingface API (model card)
[6]in the training of the module we used teacher enforcement, so there was no maximum string length; however, we did use this when generating the training data for the final module, so the final module had not seen RNN-generated inputs of over 50.

| Type | Rank | F1 | P | R |
|---|---|---|---|---|
| Dev (English) | 27/33 | 0.55 | 0.47 | 0.66 |
| Test English | 25/33 | 0.56 | 0.48 | 0.67 |
| Test Bulgarian | 10/20 | 0.46 | 0.37 | 0.61 |
| Test North Maced. | 4/20 | 0.46 | 0.36 | 0.66 |
| Test Arabic | 3/20 | 0.43 | 0.31 | 0.70 |

Table 1: Table showing the main results of our official submissions. The rank $x/y$ shows our position $x$ and the total amount of teams that made a test submission $y$

more training data would give more robust results; as we did not do additional finetuning for the post-hoc experiments, no development set was used (see section 5). Due to the way the Semeval challenge was set up, the validation set was a dataset that was available from the beginning, whereas the gold labels for the development set only became available a couple of weeks before the test submission closed; we only used the validation set for some preliminary testing and setup, after which it was joined with the test set. All results are reported on the test data.

The modules were trained separately, but on the same data. We conducted some preliminary experiments training module 1 and 2 on 75% of the data and module 2 on the remaining 25% (random split) but this led to a drop in performance. The optimal amount of epochs for each module was decided based on plots of the average loss per epoch. Each separate module took less than 30 minutes to train on an Apple M3 8-core CPU. We used an Adam optimizer with a learning rate of 1e-3. Modules 1 and 3 were trained with a CrossEntropyLoss; module 2 with a SmoothL1Loss.

The task evaluation metric was the hierarchical-F1, calculated using hierarchical precision and recall (Kiritchenko et al., 2006). This measures gives partial points for assigning a label higher up in the hierarchy, and full points for assigning the specific technique.

## 4 Results

Table 1 shows the outcomes of our official submissions on the test and dev sets (before 1/2/2024). There was originally an issue with the Arabic gold labels; the reported scores correspond to the corrected version of the gold labels. For the English data, the test results were very much in line with the results on the dev leader board, with only 0.01 point difference in the hierarchical-F1. The results on the zero-shot test submissions were more surprising: although the F1 was (expectedly) lower

| Model description | Language | F1 | P | R |
|---|---|---|---|---|
| As test submission[7] | Eng. | 0.56 | 0.45 | **0.72** |
| | Bulg. | 0.46 | 0.35 | **0.68** |
| | Maced. | 0.45 | 0.33 | **0.70** |
| | Arabic | 0.40 | 0.28 | **0.71** |
| No NER | Eng. | 0.55 | 0.46 | 0.68 |
| | Bulg. | 0.47 | 0.37 | 0.64 |
| | Maced. | 0.46 | 0.36 | 0.64 |
| | Arabic | 0.42 | 0.30 | 0.68 |
| No preprocessing[8] | Eng. | **0.57** | 0.49 | 0.67 |
| | Bulg. | 0.47 | 0.38 | 0.62 |
| | Maced. | 0.46 | 0.36 | 0.63 |
| | Arabic | 0.42 | 0.31 | 0.64 |
| Only module 1 | Eng. | 0.53 | 0.48 | 0.6 |
| | Bulg. | 0.44 | 0.36 | 0.56 |
| | Maced. | 0.42 | 0.35 | 0.55 |
| | Arabic | 0.40 | 0.32 | 0.56 |
| Only module 2 | Eng. | 0.46 | **0.54** | 0.39 |
| | Bulg. | 0.37 | **0.41** | 0.34 |
| | Maced. | 0.32 | **0.39** | 0.27 |
| | Arabic | 0.38 | **0.34** | 0.42 |
| MLL[9] RNN = 100 | Eng. | 0.54 | 0.50 | 0.58 |
| | Bulg. | **0.48** | 0.40 | 0.61 |
| | Maced. | **0.48** | 0.38 | 0.65 |
| | Arabic | **0.42** | 0.30 | 0.68 |

Table 2: Table showing the results of subsequent experiments on the test set

than for the English data, the ranking showed that the system still generalized considerably better than most other approaches. Section 5 discusses some possible explanations and describes additional experiments aimed at shining light at this question.

## 5 Discussion

We hypothesize that two mechanism that might have contributed to the system's generalization capacity. First, the preprocessing (particularly the NER step) might have made the model more generalizable. Second, the seq2seq RNN module to learn the labels might have been particularly good at capturing the relationships between the labels

In order to investigate these hypotheses, we ran additional experiments in which we left out parts of the system to investigate what happened to the performance. The results are summarized in table 2. All of these models were trained on train, validation and development set and tested on the test set (there was no development set as no additional fine-tuning was performed). Note that this is different from the original submission, that was trained on the training and validation set, validated on the development set, and tested on the test set; the manipulations made in the post-hoc experiments should thus be compared to the results in the upper row of table 2, which is a re-run of the model as described in section 2, but trained on the 1000 more memes of

the development set.

Our first hypothesis was that the preprocessing, especially the NER, helped the system generalize better to unseen languages. However, this seems not to be the case. Taking out only the NER module let to a slight drop in performance in English, but a better performance in the other languages. A possible explanation is the non-ubiquity of the name Mark: replacing people with 'Mark' might not actually be helpful if 'Mark' is not adapted to the specific language. Skipping all preprocessing steps (other than adding the embeddings) actually improved performance for English (even though taking out only the NER led to a drop, suggesting this might actually have been a very helpful step for the English data), but made hardly any difference for the other languages when compared to the setting without NER.

Our second hypothesis was that the RNN module was especially helpful for generalization. Either module alone performed worse than the three modules combined for all languages (apart from the first module, that reached the same F1 for Arabic), so the influence of the label-generating RNN should not be overestimated. On the other hand, when comparing the performance of module 1 with the performance of module 2, we see that the difference is the same for English and Bulgarian, and bigger for both Macedonian and Arabic. This might mean towards the second module actually being a bit more important in the zero-shot setting, but more research is required.

Our full model had remarkably high recall, but low precision. Looking at the performance of modules 1 and 2 separately suggests that this is mainly due to module 1 (and, possibly, module 3, that is very similar to module 1 in architecture). This pattern is the same across languages and modifications. This is not very surprising; erroneously generating the EOS token once makes the module stop predicting labels, and given that every training instance has an EOS token, it is very common and the chance of it being produced erroneously is relatively high. Moreover, the RNN stops generating strings when the maximum string length of 50 is reached. We thus re-ran the base model (including development model) with a maximum string length of 100 for the RNN (table 2). This resulted in the best model thus far for the zero-shot setting due to improved precision, but the performance for the English test data fell marginally. This is a further indication that the RNN module is indeed crucial to the zero-shot classification.

Adding the development data (i.e. training the model on more data) seems marginally helpful for English (+0.01 point F1) but marginally unhelpful for Macedonian (-0.01 point F1) and Arabic (-0.03 point F1). If the strength of our system indeed lies in it learning the relationships between the labels of the hierarchy, it is likely that a smaller amount of data was just enough to learn this, and adding more data just makes the model overfit.

## 6 Conclusion

This paper described the system used to generate the test submissions for subtask 1 of task 4 of SemEval 2024 'multilingual detection of persuasion techniques in memes'. We proposed a system consisting of different neural modules, the most innovative of which was an RNN that was trained to generate sequences that reflect the position of the relevant labels in the hierarchy. Our model did not perform particularly well on the English data, but compared to the other teams, it generalized unexpectedly well to other languages in a zero-shot setting. We conducted some additional experiments to find out what might have contributed to this. We found that our preprocessing steps (normalization and NER) did not make the model more generalizable, but we did find some evidence that the RNN module might have played a role as hypothesized.

We see plenty of possibilities to improve on our original idea in the future. First of all, we would like to explore the performance of different types of embeddings. We found that our system as a whole had a high recall, but a low precision; however, the RNN module showed the exact opposite pattern, having high precision and low recall. Allowing the RNN output in module 3 to be longer (up to 100 tokens) partially alleviated this problem and improved performance. We hypothesize this is because the EOS token is generated too easily. Somehow raising a barrier for the module to generate the EOS token might help to improve its recall. Implementing an attention mechanism in the final module could also help aleviate this problem. Other options to explore are a NER preprocessing step that better generalizes to other languages than just replacing people with "Marks". Finally, it would be interesting to explore the capabilities of a hierarchical-path generating RNN with more sophisticated layers (GRU or LSTM), or combined with a convolutional model.

## Funding and acknowledgements

This research was carried out in the context of the HYBRIDS project. This project has received funding from the European Union´s Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 101073351. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

This research was carried out at the Fondazione Bruno Kessler (Trento, Italy) and the CiTIUS (Santiago de Compostela, Spain). A special thanks to dr. Sara Tonelli and dr. Marcos García González for supervising this project, Erik Bran Marino for his comments on the paper draft, and Rafael Frade for the valuable feedback and support. Finally I would like to thank the anonymous reviewers for their time, effort and important and helpful insights.

## References

Jasmin Bogatinovski, Ljupčo Todorovski, Sašo Džeroski, and Dragi Kocev. 2022. Comprehensive comparative study of multi-label classification methods. *Expert Systems with Applications*, 203:1–18.

Xiaolong Chen, Jieren Cheng, Zhixin Rong, Wenghang Xu, Shuai Hua, and Zhu Tang. 2023. Multi-label text classification based on improved seq2seq. In *International Conference on Computer Engineering and Networks*, pages 439–446. Springer.

Ziheng Chen and Jiangtao Ren. 2021. Multi-label text classification with latent word-wise label information. *Applied Intelligence*, 51:966–979.

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.

Eleonora Giunchiglia and Thomas Lukasiewicz. 2020. Coherent hierarchical multi-label classification networks. *Advances in neural information processing systems*, 33:9662–9673.

Chenyang Huang, Amine Trabelsi, Xuebin Qin, Nawshad Farruque, Lili Mou, and Osmar R Zaiane. 2021. Seq2emo: A sequence to multi-label emotion classification model. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 4717–4724.

Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed. 2023. A review of deep transfer learning and recent advancements. *Technologies*, 11(2):40.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7-9, 2006. Proceedings 19*, pages 395–406. Springer.

Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. 2021. General multi-label image classification with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16478–16488.

Ahmed Omar, Tarek M Mahmoud, Tarek Abd-El-Hafeez, and Ahmed Mahfouz. 2021. Multi-label arabic text classification in online social networks. *Information Systems*, 100:1–18.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Miguel Romero, Felipe Kenji Nakano, Jorge Finke, Camilo Rocha, and Celine Vens. 2023. Leveraging class hierarchy for detecting missing annotations on hierarchical multi-label classification. *Computers in Biology and Medicine*, 152:106423.

Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. 2021. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118:107965.

Ran Wang, Robert Ridley, Weiguang Qu, Xinyu Dai, et al. 2021a. A novel reasoning mechanism for multi-label text classification. *Information Processing & Management*, 58(2):102441.

Wei Wang, QiuYing Dai, Fang Li, Yi Xiong, and Dong-Qing Wei. 2021b. Mlcdforest: multi-label classification with deep forest in disease prediction for long non-coding rnas. *Briefings in Bioinformatics*, 22(3):1–11.

Yaoqiang Xiao, Yi Li, Jin Yuan, Songrui Guo, Yi Xiao, and Zhiyong Li. 2021. History-based attention in seq2seq model for multi-label text classification. *Knowledge-Based Systems*, 224:107094.