Combining Language Models and Linguistic Information to Label Entities in Memes

Pranaydeep Singh, Aaron Maladry, Els Lefever
LT3, Language and Translation Technology Team, Ghent University, Belgium
Groot-Brittanniëlaan 45, 9000 Ghent, Belgium
firstname.lastname@ugent.be

Abstract

This paper describes the system we developed for the shared task "Hero, Villain and Victim: Dissecting harmful memes for Semantic role labeling of entities" organized in the framework of the Second Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation (Constraint 2022). We present an ensemble approach combining transformer-based models and linguistic information, such as the presence of irony and implicit sentiment associated to the target named entities. The ensemble system obtains promising classification scores, with a macro F-score of 55%, resulting in a third place finish in the competition.

1 Introduction

The exponential growth of social media such as Twitter, Facebook or Youtube has created a variety of novel ways to communicate. This daily exposure to other users' opinions and comments has become a constant in many people's lives. Unfortunately, this new way of freely communicating online has also given a forum to people who want to denigrate others because of their race, color, gender, sexual orientation, religion, etc., or to spread fake news and disinformation. The automatic processing of this user generated text by means of Natural Language Processing (NLP) techniques may contribute to an effective analysis of public opinion, but also to the automatic detection of this harmful online content.

One very popular mode of expression on social media today are internet memes. Memes are often used for entertainment purposes, but they are also used for online trolling, because of their potential for spreading provocative and attention-grabbing humor (Leaver, 2013). They have been described both as speech acts (Grundlingh, 2018) and performative acts, involving a conscious decision to either support or reject an ongoing social discourse (Gal et al., 2016). Their multi-modal nature, composed of a mixture of text and image, makes them a very challenging research object for automatic analysis. Research has already been proposed to automatically process harmful memes in various downstream tasks. A related shared task was proposed by Kiela et al. (2020), who organized the hateful memes challenge, where systems were developed to detect hate speech in multimodal memes. Most systems participating to the task applied fine-tuning of state-of-the-art transformer methods, such as supervised multimodal bitransformers (Kiela et al., 2022), ViLBERT (Lu et al., 2019) and VisualBERT (Li et al.) to classify memes as being hateful or not.

This paper presents our system developed to classify entities as *hero*, *villain*, *victim* or *other*, in memes about two controversial topics provoking a lot of hate speech and disinformation, namely the presidential election in the US and the COVID-19 pandemic spreading. To tackle the task, we incorporated both transformer-based embeddings as well as linguistic information (implicit entity connotations and irony detection labels) into our classifier.

The remainder of this paper is organized as follows. Section 2 introduces the shared task and data sets, whereas Section 3 describes the information sources and ensemble system we developed to label named entities in memes. Section 4 lists the experimental results and provides a detailed analysis and discussion. Section 5 ends with concluding remarks and indications for future research.

2 Shared Task and Data

The research described in this paper was carried out in the framework of the Constraint 2020 shared task: *Hero, Villain and Victim: Dissecting harmful memes for Semantic role labeling of entities* (Sharma et al., 2022). Given a meme and an entity, systems have to determine the role of the

Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations, pages 35 - 42
May 27, 2022 ©2022 Association for Computational Linguistics

	Villain	Hero	Victim	Other	Total nr of entities
COVID-19 train memes					
2700 memes	662	190	360	6022	7234 (1927 unique)
Politics train memes					
2852 memes	1765	285	550	7680	10280 (2798 unique)
Total train memes					
5552 memes	2427 (14%)	475 (3%)	910 (5%)	13702 (78%)	17514 (4398 unique)
Held-out test memes					
718 memes	350 (14%)	52 (2%)	114 (5%)	1917 (79%)	2433 (1103 unique)

Table 1: Statistics of the training and test data set, showing the number of entities per class, and the unique number of entities per data partition.

entity in the meme, namely:

- *hero*: “The entity is presented in a positive light. Glorified for their actions conveyed via the meme or gathered from background context”
- *villain*: “The entity is portrayed negatively, e.g., in an association with adverse traits like wickedness, cruelty, hypocrisy, etc.”
- *victim*: “The entity is portrayed as suffering the negative impact of someone else’s actions or conveyed implicitly within the meme.”
- *other*: “The entity is not a hero, a villain, or a victim.”

The task is conceived as a multi-class classification task, which has to be analyzed from the meme author’s perspective.

2.1 Training and Test Data

The task organizers provided training data for two controversial topics triggering a lot of hostile social media posts, and memes in particular, viz. the presidential election and COVID-19 pandemic. Table 1 shows the statistics of the training and held-out test data. As can be noticed, the data set is very skewed towards the “other” category (78% of the training and 79% of the test entities). It is also interesting to mention that out of the 1103 unique test entities, only 542 entities also appeared in the training data.

The data was provided in the following json format, containing the OCR’ed text from the meme, the file name of the corresponding meme, and a list of gold entities per category:

{“OCR": "IF PROPERLY FITTED, ONE MASK CAN\n SAVE MANY THOUSANDS OF LIVES\n Dr. Fauci\nXESH\nHE WH\nWASE\n”, “image”: “covid_memes_1797.png”, “hero”: [“dr. anthony fauci”], “villain”: [“donald trump”], “victim”: [], “other”: [“mask”] }

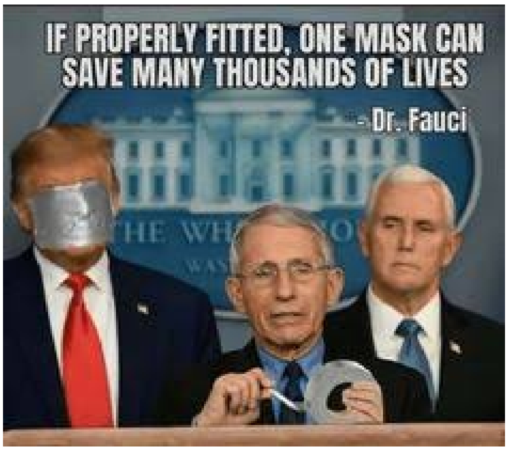
IF PROPERLY FITTED, ONE MASK CAN
SAVE MANY THOUSANDS OF LIVES
- Dr. Fauci

Figure 1: covid_memes_1797.png

3 System Description

We approached the meme entity labeling task as a multi-class classification task, where a category is predicted for all entities occurring in the meme. To this end, an ensemble classifier is built combining probability scores output by various transformer-based language models and linguistic information assigning implicit sentiment to the entities and detecting irony in the meme text. We first give an overview of all different information sources in-

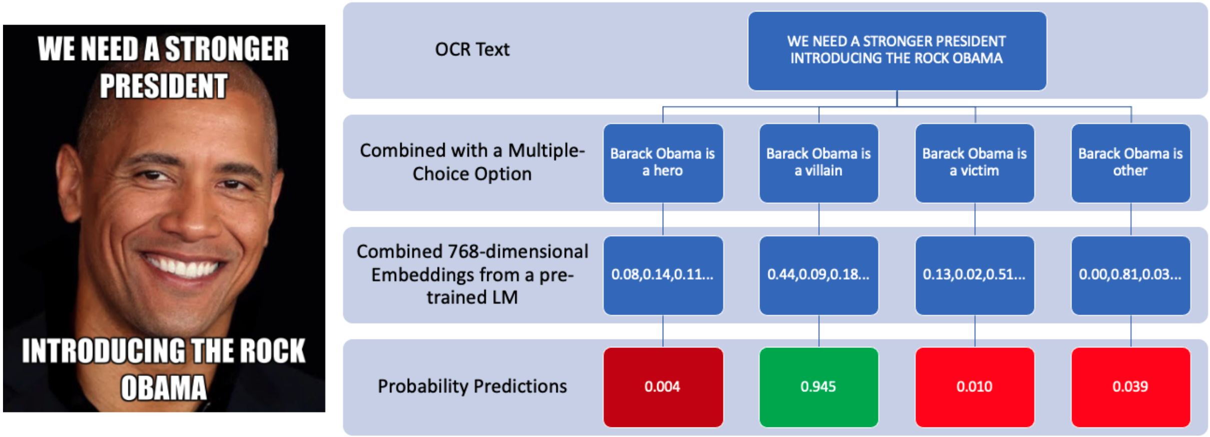
WE NEED A STRONGER PRESIDENT
INTRODUCING THE ROCK OBAMA
OCR Text
WE NEED A STRONGER PRESIDENT INTRODUCING THE ROCK OBAMA
Combined with a Multiple-Choice Option
Barack Obama is a hero
Barack Obama is a villain
Barack Obama is a victim
Barack Obama is other
Combined 768-dimensional Embeddings from a pre-trained LM
0.08,0.14,0.11...
0.44,0.09,0.18...
0.13,0.02,0.51...
0.00,0.81,0.03...
Probability Predictions
0.004
0.945
0.010
0.039

Figure 2: Illustration of the MCQA setup and features obtained for the transformer-based language models.

corporated in the feature vector (Section 3.1), and then describe the ensemble method combining the various information sources into a feature vector for classification (Section 3.2).

3.1 Information Sources

3.1.1 Transformer-based Language Models

The information used for our first feature group are similarity probabilities per class output by state-of-the-art transformer-based language models. As the target entities do not (always) occur in the OCR'ed meme text (for example, "Donald Trump" is an entity not present in the text in Figure 1), we had to find a different way to fine-tune the pre-trained language models for labeling the entities. To tackle this issue, we recast the labeling task as a multiple choice QA task (MCQA), where the various questions are formulated as "<entity> is a hero", "<entity> is a vilain", etc. The model then appends the question (OCR'ed meme text) to each option individually, and computes a probability output for the similarity.

Three different transformer-based pre-trained language models were fine-tuned for the task, applying different transformer architectures, namely BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) and pre-trained on different types of data: (1) twitter-base-roberta, (2) bert-tweet, and (3) COVID-bert.

twitter-base-roberta (Barbieri et al., 2020) is trained on 58M tweets and is a language model applying a RoBERTa architecture. While Twitter data is already closer to meme text than the standard Wikipedia and Common Crawl text, the tweets collected for training this language model are quite a bit older than our shared task data set.

COVID-bert (Müller et al., 2020) is trained on a corpus of 160M more recent tweets (spanning the first half of 2019) about the corona virus. The content of the tweets is, however, very related to the content of the shared task data, as they contain covid-related key words.

bert-tweet (Nguyen et al., 2020) uses similar pre-training data to twitter-base-roberta but is a larger architecture with significantly increased and recent pre-training data. The large RoBERTa architecture was trained on 850M English Tweets, containing 845M Tweets streamed from 01/2012 to 08/2019 and 5M Tweets related to the COVID-19 pandemic.

Each pre-trained language model was optimized using cross-entropy for the task of multiple-choice QA as illustrated in Figure 2. Each entity along with it's possible class, is treated as a separate multiple-choice option. The Language models were fine-tuned for 5 epochs with an LR of 1e-5, batch size of 4 per device, on 2 Tesla V100 GPUs.

3.1.2 Implicit Sentiment

The creation of the implicit sentiment feature was motivated by the assumption that entities might have a predominant connotation on Twitter. To determine the implicit sentiment of the entities, we collected 400 to 800 tweets containing each entity and combined them into a large background corpus of three million tweets. As memes and tweets both originate from social media platforms, we considered this the most reliable source for the implicit sentiment from the perspective of most users, although we recognize that meme-makers might have very different opinions about certain politicians. We analyzed the sentiment of the collected tweets with a pre-trained RoBERTa model (Heitmann et al., 2020)[1] that was pre-trained using 15 data sets across different text types, including tweets. We grouped the tweets per entity and considered the implicit sentiment of an entity to be determined by the percentages of positive, negative and neutral tweets for that entity in our background corpus. Additionally, we constructed another categorical feature reflecting the dominant implicit sentiment (positive, neutral or negative). This way, we ended up with four *implicit sentiment* features: the distribution values for positive, negative and neutral tweets in the background corpus and the dominant sentiment for that target entity based on those values. These features were finally combined with the output of the BERT question-answering systems into the ensemble model.

3.1.3 Irony Detection

As we assume that a lot of memes contain figurative language, and irony in particular, we modeled a second linguistic feature by performing irony detection on the OCR text. To detect irony, we used a pre-trained RoBERTa model (Barbieri et al., 2020)[2], which contains the RobBERTa-base model and was fine-tuned using the SemEval 2018 data set for Irony Detection in English tweets (Van Hee et al., 2018). The value of the resulting feature is the probability score for the irony label (between 0 and 1).

In hindsight, we think most of the irony did not occur inside the OCR text but is expressed in a multi-modal way between the image and the text. This was confirmed by the experimental results, as the feature for irony detection inside the OCR text did not increase the accuracy of our system for entity classification.

3.1.4 FastText Embeddings

The final feature group we modeled is based on FastText embeddings (Bojanowski et al., 2017). As we scraped a relevant background corpus containing all target entities, we hypothesized this would also be an interesting corpus for training embeddings. Although FastText outputs static, and not contextualized embeddings, it was very popular before the transformer-based revolution in NLP, and is computationally cheap to train word vectors. First, the background corpus was tokenised using NLTK's tokenizer for tweets[3], which for instance keeps hashtags intact. FastText embedddings were then trained using the continuous-bag-of-words (cbow) model, which predicts the target word according to its context. The context here is represented as a bag of all words contained in a fixed size window around the target word. This resulted in a vocabulary of 61,871 words and 100-dimensional word vectors for the Twitter background corpus. The FastText embeddings of the entities were integrated in the feature vector as 100 separate features.

3.2 Ensemble System

We trained an ensemble system combining the results from each of the information sources listed above as features. We use the probability predictions for each class from the fine-tuned language model, an average score for each implicit sentiment (positive, negative, neutral) present in the background corpus for the respective entity, the probability score for the irony associated with the OCR text, and the 100-dimensional pre-trained FastText embeddings for the entity text (averaged for multiple tokens in an entity), resulting in a feature vector containing 108 features. We explain the construction of the feature vector with the 4 sets of features in Figure 3.

We experimented with 3 classifiers, Gradient Boosted Trees (XGB**dd**oost), Random Forest and Support Vector Machines as implemented in sklearn (Pedregosa et al., 2011). We used grid searching with 5-fold cross-validation to find the optimal hyperparameters for each classifier, and our final classifier in all cases is an SVM with an RBF Kernel, a C value of 0.1 and a gamma value of 0.01.

While experimenting with the different classifiers and features, we calculated feature importance according to the linear kernel SVM classsifier. The respective scores reflecting the contribution of the various features to solve the task are listed in Figure 4.

[1] https://huggingface.co/siebert/sentiment-roberta-large-english

[2] https://huggingface.co/cardiffnlp/twitter-roberta-base-irony

[3] https://www.nltk.org/api/nltk.tokenize.html

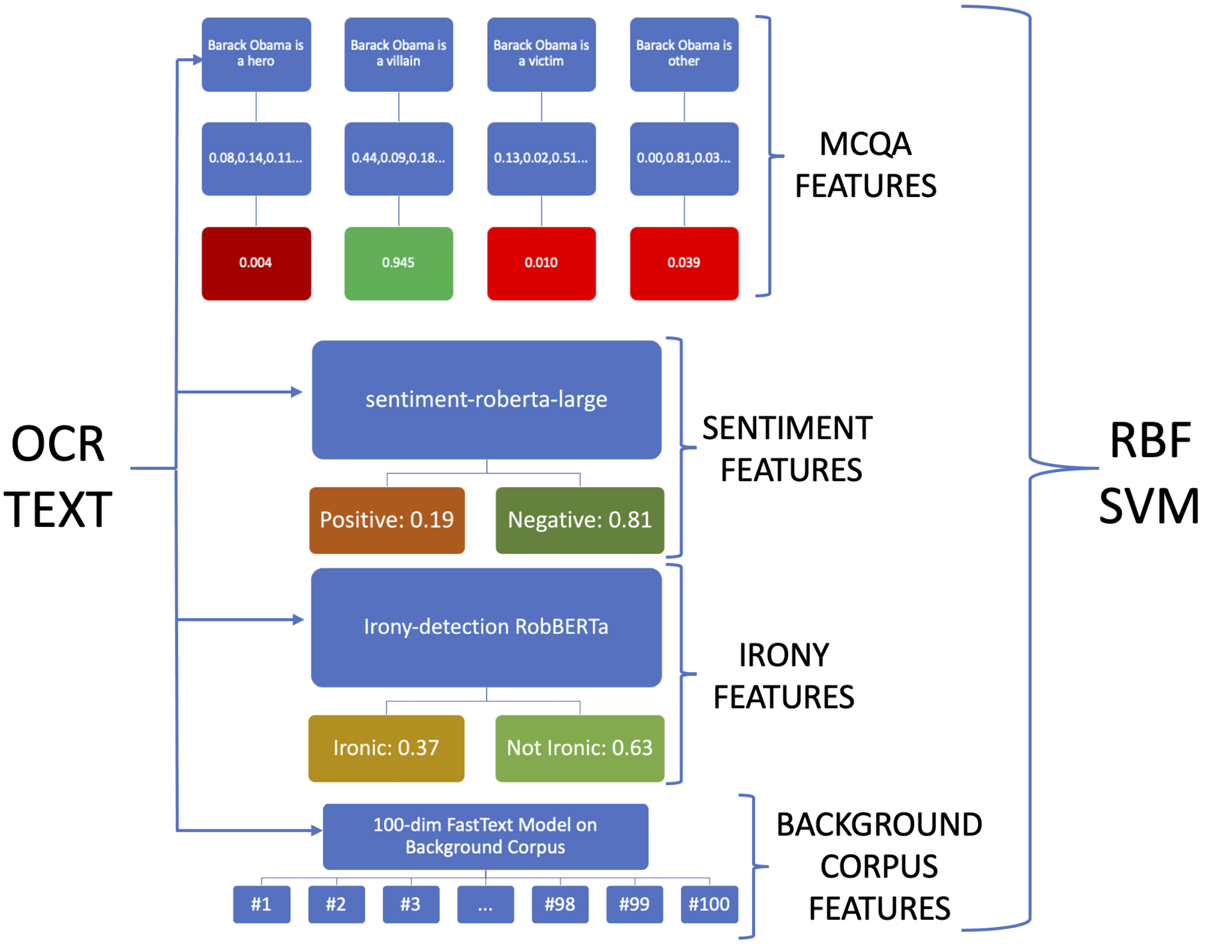

Figure 3: A visual summary of the ensemble setup and the features involved.

4 Experimental Results

A first set of experiments was carried out to assess the classification performance of the different language models. In this case, the classifier is trained and evaluated on feature vectors containing similarity scores for the four different labels. The first three lines of Table 2 show the classification scores for this multiple choice QA language model systems. It is clear from the results that the bert-tweet model performs best, resulting in a Macro F1-score of 0.5467. When adding implicit sentiment for the target entities, the score only slightly improves.

For a second set of experiments, we created an ensemble system containing various combinations of the MCQA language model probability scores per label, together with the implicit sentiment feature for the target entity. The best performing ensemble appeared to be a combination of the twitter-xlm-roberta, COVID-bert and bert-tweet similarity scores per label, together with the implicit sentiment features, resulting in the best performance scores on the held-out test set, viz. a macro F1-score of 0.5514. Combining this ensemble system with the irony detection and FastText word vector features resulted in a lower F-score (0.5495) and precision (0.5201), but in a higher recall score (0.6045).

Table 3 lists the precision, recall and F-scores per entity label for the best performing system, being the ensemble system containing the best three language model predictions together with the implicit sentiment feature. As expected, the *Other* category, which represents 78% of the training targets, performs best and the *Hero* category performs worst (only 3% of training entities), especially obtaining a very low recall of 0.27. For the other two labels, *Villain* and *Victim*, precision and recall are better balanced.

To gain more insights into the performance of the best classifier, we constructed a confusion matrix for all labels and performed an error analysis. Completely in line with the classification scores per label, we can notice in the confusion matrix (Fig-

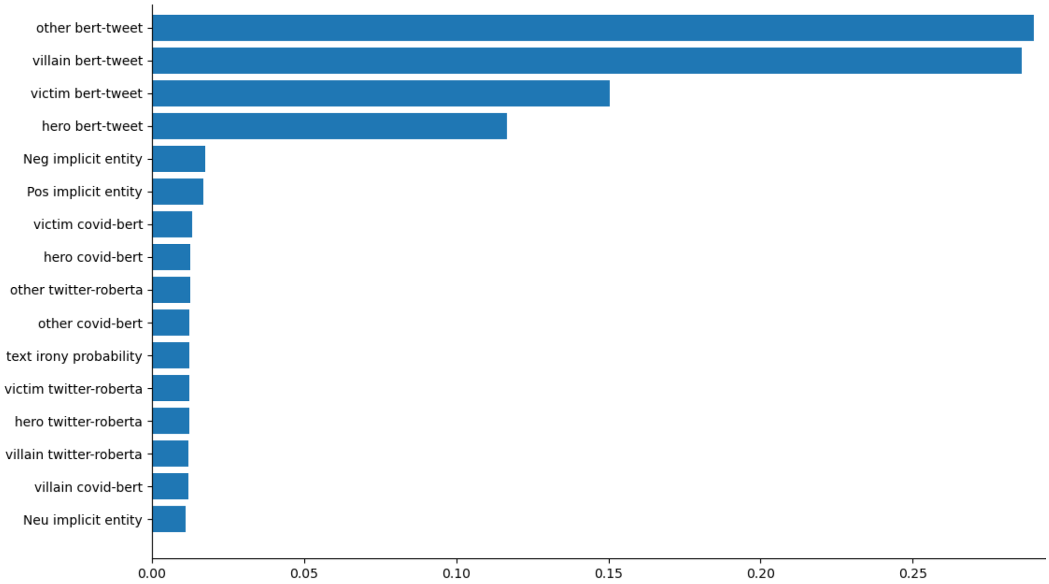

Figure 4: Feature importances of the classifier we used for our ensemble model. The features include the MCQA values per label for each of our language models, the percentages of positive, negative and neutral tweets found for the entity and the probability of the text being ironic.

Model	Macro-F1	Precision	Recall
MCQA twitter-xlm-roberta	0.3433	0.4211	0.2898
MCQA COVID-bert	0.5083	0.5188	0.4997
MCQA bert-tweet	0.5467	0.524	0.5812
MCQA bert-tweet + Sentiment	0.5471	0.5274	0.5814
MCQA ensemble + Sentiment	**0.5524**	**0.5391**	0.5725
MCQA ensemble + Sentiment + + FastText + Irony	0.5495	0.5201	**0.6045**

Table 2: Macro-averaged F1-scores, precision and recall for the various classification systems.

Label	F1-score	Precision	Recall
Hero	0.33	0.41	0.27
Villain	0.55	0.55	0.54
Victim	0.45	0.44	0.46
Other	0.89	0.88	0.89

Table 3: Classification scores (F1-score, precision, recall) for the different named entity labels.

ure 5) that most of the missed labels are wrongly predicted as "Other" (even up to 60% for the *hero* label). Another remarkable fact is that 12% of the *victim* labels are predicted as *villain*.

Apart from challenges posed by the data set itself, such as noise in the OCR text, very skewed class distribution, or spelling mistakes in the target entities [4], our error analysis revealed some other trends in wrongly predicted named entity labels.

First, it is clear that labeling entities in memes is a very hard task. Systems have to both understand the OCR text, but also correctly process the picture that sometimes contains crucial information. As we only incorporate text processing features in our ensemble system, a lot of the erroneous predictions are caused because of lacking visual information to correctly interpret the picture of the meme, as illustrated by Figure 6.

In addition, some memes require a lot of common sense or factual/news knowledge. As an example, we can refer to Figure 7, where the entity *Melania Trump* had to be labeled as "Villain", but was predicted by the system as "Other". It is impossible, however, to interpret this meme correctly without knowing that Donald Trump's wife, Melania, took center stage on the first day of the Republican National Convention, and was accused of the fact that a portion of her speech plagiarized Michelle Obama.

[4] Mistakes like "dr. dr. anthony fauci" and "valdimir puitin".

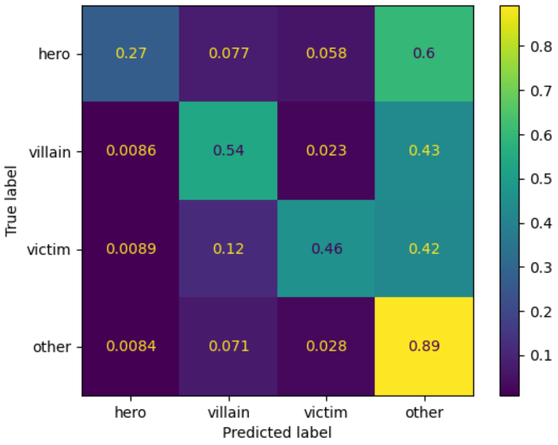

Figure 5: confusion matrix of the prediction results on the held-out test set.

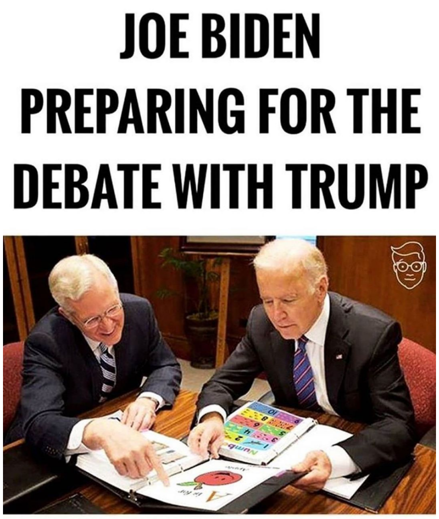

Figure 6: Meme requiring visual information features.

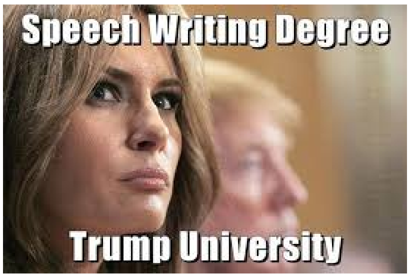

Figure 7: Meme requiring common sense/factual knowledge.

5 Conclusion

In this paper, we describe the system proposed for the Constraint 2022 shared task on labeling entities in memes as *Hero*, *Villain*, *Victim* or *Other*. To tackle the task, we built an ensemble classifier combining the output predictions of various transformer-based language models with implicit sentiment features for the target entities, irony predictions on the OCR text and FastText word vectors. The best performing system combines the predictions of three different language models with the implicit sentiment feature, obtaining a Macro F1-score of 55%. As the data set was very skewed, we obtained much better results for the "Other" class than for the other three labels. Especially for the *Hero* class, only represented by 3% of the training entities, classification appeared to be challenging (F1-score of 33%).

The analysis of the results showed there is still a lot of room for improvement. In future research, we plan to integrate visual information into our ensemble system, as it is clear that we lacked this information to properly address this multimodal task. In addition, we will investigate other ways to set up the multiple choice QA system, in order to construct better sentences containing the target entities. Finally, the system would also benefit from more semantic information, in order to model entities that are now not explicitly mentioned in the OCR text. It would, for instance, be interesting to semantically link an OCR text line talking about *Brexit* with the entity *UK Government*. This would allow to inject some common sense into the meme classification system.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Noam Gal, Limor Shifman, and Zohar Kampf. 2016. "it gets better": Internet memes and the construction of collective identity. *New media & society*, 18(8):1698–1714.

Lezandra Grundlingh. 2018. Memes as speech acts. *Social Semiotics*, 28(2):147–168.

Mark Heitmann, Christian Siebert, Jochen Hartmann, and Christina Schamp. 2020. More than a feeling: Benchmarks for sentiment analysis accuracy. *Available at SSRN 3489963*.

Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, and Davide Testuggine. 2022. Supervised multimodal bitransformers for classifying images and text. In *Proceedings of the NeurIPS 2019 Workshop on Visually Grounded Interaction and Language (VIGIL@NeurIPS'19*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS '20)*.

Tama Leaver. 2013. Olympic trolls: Mainstream memes and digital discord. *Fibreculture Journal*, 1(22):216–233.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv:1908.03557*.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS '19)*, pages 13–23.

Martin Müller, Marcel Salathé, and Per E Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Shivam Sharma, Tharun Suresh, Atharva Jitendra, Himanshi Mathur, Preslav Nakov, Md. Shad Akhtar, and Tanmoy Chakraborty. 2022. Findings of the constraint 2022 shared task on detecting the hero, the villain, and the victim in memes. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations - CONSTRAINT 2022, Collocated with ACL 2022*.

Cynthia Van Hee, Els Lefever, and Veronique Hoste. 2018. SemEval-2018 task 3 : irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50. Association for Computational Linguistics.