Kb at SemEval-2023 Task 3: On Multitask Hierarchical BERT Base Neural Network for Multi-label Persuasion Techniques Detection

Katarzyna Baraniak and Marcin Sydow Polish-Japanese Academy of Information Technology kbaraniak@pjwstk.edu.pl msyd@pjwstk.edu.pl

Abstract

This paper presents a solution for Semeval 2023 subtask3 of task3: persuasion techniques in paragraphs detection. The aim of this task is to identify all persuasion techniques in each paragraph of given news articles. We use hierarchical multitask neural networks combined with transformers. Span detection is an auxiliary task that helps in the main task: identify propaganda techniques. Our experiments show that if we change the index of BERT embedding from the first token of the whole input to the first token of the identified span it can improve the performance. Span and label detection can be performed using one network so we save data and, when data is limited, we can use more of it for training.

1 Introduction

News articles are expected to present reliable information, however quite often contain some kind of manipulation. Unconscious reader can be unable to spot all kind of persuasion that he is exposed to. Reading such news can influence a reader's point of view especially if the reader has a low level of political knowledge (Eberl et al., 2017). Automatic tools for detection of persuasion can help in media analysis and creating more objective news.

The task prepared by the organisers (Piskorski et al., 2023) covers three topics, as follows: category of news article identification, framing classification and paragraph-level persuasion techniques recognition. Our team focused on the third topic and created solution tested on all languages that have training data available: English, French, German, Italian, Polish and Russian and for additional languages with only test data: Spanish, Greek and Georgian.

Our system combines multitask learning and hierarchical neural networks. The system identifies span where the persuasion techniques appear and then passes this information to the next module that classifies it to one or more categories of persuasion techniques. We used pre-trained and fine-tuned Bert embedding as the shared input layer.

We discover that adding an auxiliary task such as span identification as the first step of training neural network may give better results than simply classifying the whole paragraph. Moreover, using one neural network for that approach makes solving the task possible even for limited data.

The code of our solution is available at https://github.com/Katarzynaa/ persuasion_detection

2 Background

The persuasion technique identification problem has been addressed before as a problem of span and technique classification. There exist several works on solving the persuasion techniques detection problem, for example: as a problem of multimodal image and text input (Dimitrov et al., 2021) or as problem of span and technique classification (Da San Martino et al., 2020). In our case it is the multi-label classification task, the authors of the task are expected to classify all persuasion techniques that appear in every paragraph of a news article. The input data is a single paragraph, the output is a list of detected techniques. The data is provided in three folders: "train", "dev" and "test". Each folder contains files with articles. Each article is divided into paragraphs separated by an additional empty line. For each paragraph of each article in "train" and "dev" sets organisers provide the lists of labels of persuasion techniques in separate files. Labels of test data are unknown. As an additional information authors provide for each paragraph the list of spans, start and end character index, and its class.

There are 23 classes with a distribution that is not balanced (far from the uniform one). These classes are presented in table 1 One can find a detailed description in the task description paper.

Technique	N. test	N. train
Doubt	187	518
Whataboutism	2	16
Appeal to Hypocr.	8	40
Causal Oversimp.	24	213
Appeal to Author.	28	154
Guilt by Associat.	4	59
Slogans	28	153
Flag Waving	96	287
Loaded Lang.	483	1809
Red Herring	19	44
False DilNoCh.	63	122
App. to Popular.	34	15
Convers. Killer	25	91
Name CallLab.	250	979
A.to Fear-Prejud.	137	310
ExaggeratMini.	115	466
Repetition	141	544
Straw Man	9	15
ObfVagConf.	13	8

Table 1: Number of labels in English dataset

2.1 Multitaskand and hierarchical Networks for persuasion techniques identification

Our solution is based on multitask networks. This kind of network shares the same part or full architecture to solve several tasks being trained at once. Much research evidence demonstrates that proper choosing of auxiliary task may help to get better results at the main one (e.g. (Bjerva, 2017)).

Multi-task learning is mostly used for emotion and sentiment analysis (Zhang et al., 2022), aspect based sentiment analysis (He et al., 2019) or named entity recognition, etc.

Hierarchical networks are formed as an acyclic graph, which means that the tasks are learned by the networks' modules in some order. The results of previous modules influence the next modules. Hierarchical multitask approach has many forms and applications in NLP, for example: embedding learning (Sanh et al., 2019) or aspect-based sentiment analysis (Wang et al., 2021), etc. In our solution we created a network that solves two tasks: span identification and persuasion techniques identification. Both share the same input of Bert embedding but have separate layers for the classification part. The second one uses the results of the first one. Span identification can be treated as sequence tag classification. We predict the span where the persuasion technique is present. For the second task we used the predicted span to identify the technique used in that span. If more than one span is identified we predicted techniques based on the first index. One span can represent many persuasion techniques.

In contrast to the network presented in (Ju-

rkiewicz et al., 2020) our main aim is only persuasion technique detection, we do not use special tokens and we use one multitask network.

We also experimented with other modifications or variants of network architectures. For example, we tried to add auxiliary task of POS tagging, adding general sentiment based on "Vader" approach (Hutto and Gilbert, 2014), add entity-level sentiment trained on the SEN benchmark dataset (Baraniak and Sydow, 2021) however no improvement was observed so details are not included.

In our preliminary approach we tried to translate articles between languages to get more data but no improvement was observed. That can be caused by the fact that specific persuasion techniques for a given language are not easy to translate to other language or the fact that a basic translator omits persuasion techniques and translates it in more neutral words.

3 System overview

Our system proceeds in the following general steps:

1. Use pre-trained model to continue pretraining using masked language model task on provided news articles data from all subtasks

2. Train the multitask hierarchical model on specific tasks: span detection and multilabel classification

3. Evaluate model on the devset

4. Train model on joint trainset and devset

For the step 1 we used standard code provided by huggingface¹. Step 2 uses our implementation of a multitask hierarchical BERT based network that is described in the next subsection.

One approach is used for all languages that contain train and dev set but models are trained separately.

For languages that do not have train and dev set we translate the articles from Polish to the required language. For these languages we use simple BERT model for sequence classification. We do not use our model as we believe it is hard to find exact spans after translation.

3.1 Model architecture

General schema of the model architecture is presented in Figure 1.

The first layer of a model is a BERT layer. It takes tokenized input paragraph and calculates embeddings. Next we add dropout layer. Then it is

```
<sup>1</sup>https://huggingface.co/
```



Figure 1: Proposed model architecture. Bert embedding layer is shared between two linear layers. First linear layer identifies the first token index (dark blue arrow) of persuasion text span. Then its embedding of the first token of persuasion span is passed to the second linear layer that performs multilabel classification. The index of span embedding even for the same sample may change during training.

followed by the first linear layer responsible for tag classification for span identification. We use softmax as activation function of this layer. It classifies each token as I or O.

We take the index of the first token of a span and we use it to find the BERT embedding from this token from the previous layer. Then this embedding is passed to the second linear layer. If no span is identified the layer takes the first token of BERT embedding. The second linear layer is followed by sigmoidal activation function. It has as many outputs as the number of classes and determines whether the sample belongs to each class or not (it can activate for any class).

Loss function for our model is a sum of two:

$$Loss = 0.5 * Loss_1 + Loss_2 \tag{1}$$

The $Loss_1$ is cross entropy that calculates the loss for span identification and is used without any modification.

$$Loss1_{n} = -[y_{n}logx_{n} + (1 - y_{n}) * log(1 - x_{n})]$$
(2)

where y_n is the correct answer and x_n is the prediction for the nth batch. The $Loss_2$ is used for a multi-label classification and is binary cross entropy calculated for each label with added weight p_c :

$$Loss2_{n,c} = -[p_c y_n log x_n + (1 - y_n) * log(1 - x_n)]$$
(3)

We add weight p_c because of unbalanced number of classes. Weight of a class c is a proportion of samples from other classes to the number of samples from class *c* calculated as follows:

$$p_c = \frac{n_{all} - n_c}{n_c} \tag{4}$$

where n_{all} is the number of all samples in the dataset and n_c of the current class.

4 Experimental Setup

For final evaluation we use train file for training and test file for evaluation. After choosing the right parameters we train the whole model on both datasets.

We used random search for hyper-parameter tuning. For final model we used hyper-parameters as follows. Maximum length of a sequence is 256, as it speeds up the computation and we did not see much improvement when using longer sequences. Train batch size is 8 as it is mostly limited by the machines that were available. Maximum number of epochs is 30 and we saved the best model on dev set according to the micro-f1 metric. Learning rate was set to 1e-05 and max gradient norm was set to 10. Dropout before both linear layers is set to 0.1. The network is trained using AdamW optimization algorithm with weight decay equal to 0.01.

4.1 Data preprocessing

Spans are provided as a list of starting and ending indexes of characters. We converted them to list of I and O tags where each tag corresponds to one word. Tag O means that the word does not contain persuasion and I means that word belongs to a persuasion span. First we identify which parts are between start and end index and then we split into words using the "spacy" library².

When preparing the input data to the BERT model, transformers have their own tokenizer and it may split the words into smaller tokens. In such case each token has the same tag as the original word.

All labels were provided as a list. We convert them to multi-hot encoding vectors. Pre-trained models from table 2 are used in our retraining. All of them come from the huggingface³. We use all sets from all the three tasks to continue pre-training our BERT model for a particular language. We use masked language model for that.

Models were evaluated by $micro - F_1$ in a first place and then $macro - F_1$.

²https://spacy.io/

³https://huggingface.co/

Language	model name
English	bert-base-cased
Polish	dkleczek/bert-base-polish-uncased-v1
French	dbmdz/bert-base-french-europeana-cased
Italian	dbmdz/bert-base-italian-uncased
Russian	DeepPavlov/bert-base-bg-cs-pl-ru-cased
German	dbmdz/bert-base-german-uncased
Spanish	dccuchile/bert-base-spanish-wwm-uncased
Greek	nlpaueb/bert-base-greek-uncased-v1
Georgian	bert-base-multilingual-cased

Table 2: Transformer models used for continuing pre-training.

5 Results

The results on the official test set are presented in table 4. Our system for all languages performs better than the baseline. Interestingly, models that have no train and dev set such as Spanish, Greek or Georgian still perform better than baseline but much worse than other languages. That means we can achieve better performance using translation but it is never good enough. The reason could be that translation tries to preserve the meaning but may skip or change the type of persuasion. Also the techniques for different languages may vary.

All models on the dev set perform much better than the baseline (Figure 3). According to micro-f1, the results for 3 random runs of model are stable, the standard deviation is rather low. According to macro-f1, the standard deviation is slightly higher, that means one can observe differences between dev set results and test set results. The reason may be that the data for test set may come from different topics and time. The system learns the techniques that are more specific for train and dev set.

5.1 Error analysis

We analysed which classes in the English devset are the easiest/hardest to be recognized (Table 5). The most frequent class ("loaded language") gets a high f-measure value, but the highest one is achieved by the "Guilt by Association" class which has only 4 and 59 observations in the test and train sets, respectively, which means it is probably the easiest class to detect. Classes "Loaded Language" and "Name-Calling Labeling" get high recall but lower precision, which means they are often wrongly detected. "False Dillema-No Choice" gets much higher precision than recall which means it is precisely recognized. There are a few classes that are not recognized at all and most of them have the low number of samples.

	Our model		baseline	
Language	r/t	f-micro	f-micro	
Polish	11/20	0.31427	0.17928	
French	11/20	0.36246	0.24014	
Italian	15/20	0.39874	0.39719	
Russian	12/19	0.25289	0.20722	
German	12/20	0.37264	0.31667	
English	23/23	0.06022	0.19517	
	12*	0.30113*		
Spanish	13/17	0.24490	0.24843	
Greek	12/16	0.15021	0.08831	
Georgian	13/16	0.15017	0.13793	

Table 3: Results on the final test sets. For English language we present post-evaluation result marked as *, as the reason of the lower score on official evaluation is a wrong file uploaded. The rank with * is calculated based on official leader board. r/t-rank/total number of participants, baseline - official baseline, svm model with unigrams and bigrams as input

Some classes may require another approach, like adding broader context. For example "Red Herring" is when someone introduces irrelevant information, what may be hard to detect based on a single sentence like *He died there*. what was classified wrongly as "Repetition" or "*Melania paired the mid-length half price frock with Christian Loubotin heels*" what was classified as "Loaded language" and "Name-Calling-Labelling". Both cases are hard to be recognized without the context. For example, the second is suited to an article about fashion but not about politics.

The system was wrong also about the "Conversation Killer" technique which is often a short and rather obvious statement: "Everybody knows it." or hidden in some long paragraph "How about sorting that stuff out instead of politicizing something that should be fun for everyone? How many times does it have to be said".

We noticed that sometimes a broader context not only from the article but also from the world of politics is necessary to correctly recognise a technique. For example, the paragraph containing "Appeal to Hypocrisy" *But he didn't mention Mueller for the rest of the day.* and *Of course, Sir Kim would have had plenty of targets had he decided to pass judgement on the present incumbent of the White House.* are not easy to be classified based only on what is given in the sentence.

Only 129 paragraphs were predicted correctly

	Our r	nodel	BERT		baseline	
Language	f-micro	f-macro	f-micro	f-macro	f-micro	f-macro
English	0.4080 ± 0.001	0.1604 ± 0.030	0.3811±0.012	$0.1454{\pm}0.003$	0.16125	0.21735
Polish	0.3672 ± 0.006	0.2113±0.023	0.3624 ± 0.002	0.2181 ± 0.008	0.12524	0.05673
French	0.4157 ± 0.009	0.2825 ± 0.014	$0.3859 {\pm} 0.005$	0.2849 ± 0.006	0.29285	0.13484
Italian	0.4351 ± 0.020	0.2207 ± 0.005	0.4022 ± 0.028	0.2051 ± 0.006	0.38918	0.10385
Russian	0.4460 ± 0.004	$0.1598 {\pm} 0.020$	$0.3997 {\pm} 0.02$	0.1469 ± 0.02	0.25316	0.04284
German	0.4120 ± 0.001	0.2373±0.016	0.3964 ± 0.001	$0.2359 {\pm} 0.01$	0.33116	0.10016

Table 4: Mean scores achieved on the dev set for all language. Our model for each language was run 3 times. For baseline we get the results from the leader board

Technique	precision	recall	f1
Doubt	0.29	0.25	0.27
Whataboutism	0.00	0.00	0.00
Appeal to Hypocr.	0.00	0.00	0.00
Causal Oversimp.	0.06	0.08	0.07
Appeal to Author.	0.10	0.04	0.05
Guilt by Associat.	0.60	0.75	0.67
Slogans	0.26	0.25	0.25
Flag Waving	0.46	0.50	0.48
Loaded Lang.	0.49	0.89	0.63
Red Herring	0.00	0.00	0.00
False DilNoCh.	0.30	0.05	0.08
App. to Popular.	0.00	0.00	0.00
Convers. Killer	0.00	0.00	0.00
Name CallLab.	0.39	0.70	0.50
A.to Fear-Prejud.	0.26	0.15	0.19
ExaggeratMini.	0.19	0.38	0.26
Repetition	0.19	0.04	0.06
Straw Man	0.00	0.00	0.00
ObfVagConf.	0.00	0.00	0.00

Table 5: Scores for each class achieved by our model on English devset . Prec, rec, f1 are precision recall and f1-measure respectively.

from the English devset (all true labels are correctly recognised and no other labels). Most of them have one or two labels.

6 Conclusion

We discovered that simple change of the index in Bert embedding may help to improve the persuasion classification. Moreover, we are able to identify spans and perform classification on limited data using the described networks. Our system works better than classic BERT for sequence classification but still needs some improvements.

The next step would be to check another language models, that may be more powerful for this task. One could improve so that all the detected spans are included in classification. If more data is available, we can use separate network for spans and for classification. As a future work we should also extend the error analysis and find an approach for finding classes that are poorly recognized.

We believe that persuasion techniques are strongly diverse and hard to be detected by one model. We think that trying ensemble models could improve the results.

References

- Katarzyna Baraniak and Marcin Sydow. 2021. A dataset for sentiment analysis of entities in news headlines (sen). *Procedia Computer Science*, 192:3627–3636. Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 25th International Conference KES2021.
- Johannes Bjerva. 2017. Will my auxiliary tagging task help? estimating auxiliary tasks effectivity in multi-task learning. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 216–220.
- Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. Semeval-2020 task 11: Detection of propaganda techniques in news articles. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414.
- Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 task 6: Detection of persuasion techniques in texts and images. In *Proceedings of the* 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 70–98, Online. Association for Computational Linguistics.
- Jakob-Moritz Eberl, Hajo G Boomgaarden, and Markus Wagner. 2017. One bias fits all? three types of media bias and their effects on party preferences. *Communication Research*, 44(8):1125–1148.
- Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguis*-

tics, pages 504–515, Florence, Italy. Association for Computational Linguistics.

- Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.
- Dawid Jurkiewicz, Łukasz Borchmann, Izabela Kosmala, and Filip Graliński. 2020. ApplicaAI at SemEval-2020 task 11: On RoBERTa-CRF, span CLS and whether self-training helps them. In Proceedings of the Fourteenth Workshop on Semantic Evaluation, pages 1415–1424, Barcelona (online). International Committee for Computational Linguistics.
- Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. Semeval-2023 task 3: Detecting the category, the framing, and the persuasion techniques in online news in a multi-lingual setup. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, SemEval 2023, Toronto, Canada.
- Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6949–6956.
- Xinyi Wang, Guangluan Xu, Zequn Zhang, Li Jin, and Xian Sun. 2021. End-to-end aspect-based sentiment analysis with hierarchical multi-task learning. *Neurocomputing*, 455:178–188.
- T Zhang, X Gong, and CLP Chen. 2022. Bmtnet: Broad multitask transformer network for sentiment analysis. *IEEE Transactions on Cybernetics*, 52(7):6232–6243.