FramingFreaks at SemEval-2023 Task 3: Detecting the Category and the Framing of Texts as Subword Units with Traditional Machine Learning

Rosina Baumann
University of Tübingen
rosina.baumann@student.uni-tuebingen.de
Sabrina Deisenhofer
University of Tübingen
sabrina.deisenhofer@student.uni-tuebingen.de

Abstract

This paper describes our participation as team FramingFreaks in the SemEval-2023 task 3 "Category and Framing Predictions in online news in a multi-lingual setup." We participated in subtasks 1 and 2. Our approach was to classify texts by splitting them into subwords to reduce the feature set size and then using these tokens as input in Support Vector Machine (SVM) or logistic regression classifiers. Our results are similar to the baseline results.

1 Introduction

Fake news and false arguments are everywhere. In Twitter posts or news articles, people use persuasion techniques to influence the opinion of other people. For this reason, it is important to detect the framing and persuasion techniques in texts automatically. This was the aim of the third shared task of SemEval 2023 (Piskorski et al., 2023), in which we took part as team FramingFreaks. This task focuses on the detection of non-neutral texts (first subtask), evaluating the frame of the argumentation on the article level (second subtask), and finding out which techniques were used on the sentence level (third subtask). The data consists of online news and Twitter posts in English, French, German, Italian, Polish, and Russian. For the test set the organizers of the task proposed a few-shot learning task where our systems had to generalize to sources of other languages, namely Spanish, Greek, and Georgian.

We participated in the first and second subtasks in all languages. We did not tackle the third subtask. The focus of our contribution is on computationally simple and efficient models. We use 'traditional' machine learning models, like SVM. However, unlike word or character-based features that are commonly used with these models, we use subword units to further reduce the feature set size.

Besides their computational efficiency, the traditional classifiers based on bag-of-words (or n-grams) features have several other interesting properties. For example, unlike recent deep-learning models, they can process sequences of arbitrary length, the computational cost comes mainly comes from the size of the feature set. Convex optimization methods used in (some of) these models make training and tuning these models more straightforward. Furthermore, in some cases, they may work as well or even better than deep learning models, especially when information from pre-training is not available (see, for example, Çöltekin and Rama, 2018; Piskorski and Jacquet, 2020).

However, the bag-of-words features may typically result in sparse features. This may particularly be bad for morphologically complex languages as many words will be observed only a few times, and the relation between the morphologically-related words cannot be used by the classifier. A common alternative is to use character n-grams (e.g., Ifrim et al., 2008; Escalante et al., 2011; Han et al., 2013; Kulmizev et al., 2017), often in combination with word n-grams. The character n-gram features, on the other hand, may result in very large feature sets.

Inspired by recent deep learning approaches that use subwords as input units, we use subword n-grams as features used by traditional machine learning models. Subwords may allow models to learn from smaller, meaningful units, by reducing the sparsity and the number of features at the same time. This should reduce the vocabulary size because parts of words should be more frequent than the word itself. Especially in languages with rich morphology, since the stem often does not change. We release our code on GitHub.[1]

2 Background

2.1 Data Description

The input for the task was articles and Twitter posts with a number of tokens ranging from 100 to 10000

[1] https://github.com/cicl-iscl/FramingDetection

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 922–926

tokens. The output of the first subtask was either 'opinion', 'satire', or 'reporting'. Thereby opinion pieces were the majority class in most languages. In the second subtask, the same texts were classified in a multi-class classification of 23 framings, for example, 'Economic', 'Cultural identity', 'Legality', or 'Morality'. Examples can be found in the annotation guidelines.[2] We made predictions for all languages that were given in the task.

2.2 Data Description

We did some statistical analysis of the data to better understand the fallacies and problems of our model. First, we compared the distributions of the labels in subtask 1 in the training, and the dev set (Figure 2, 1). In the training set there, the majority proportion were opinion pieces and in the dev set, the majority class was reporting.

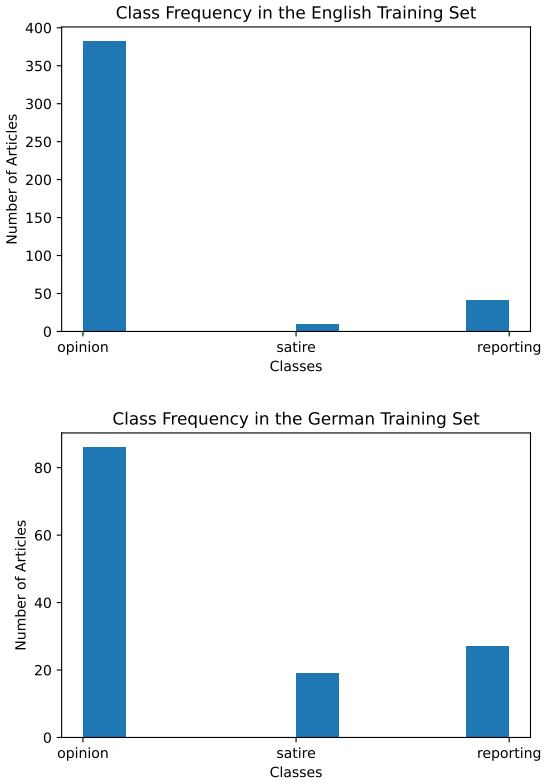

Figure 1: The distribution of the labels in the English and German training dataset.

In the German dataset, the class frequencies in the training and the dev set are similar (Figure 1, 2). These distribution differences could make the predictions harder.

[2] https://propaganda.math.unipd.it/semeval2023task3/data/annotation_guidelines.pdf

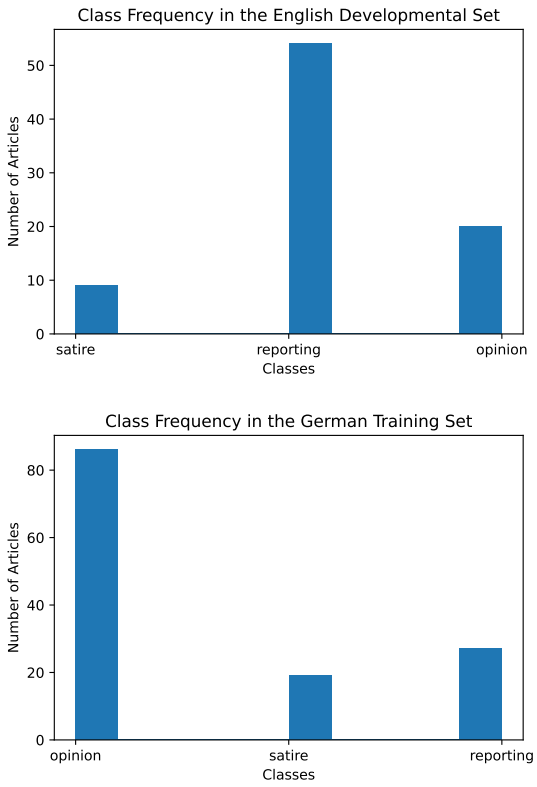

Figure 2: The distribution of the labels in the English and German dev dataset.

An additional challenge could be the length of the texts. Figure 3 shows the text length of our tokenized texts. For transformer models, this could be a problem but our system does not depend on the text length.

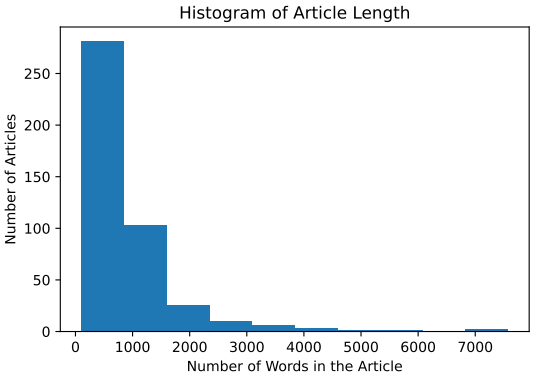

Figure 3: The text length of the tokenized text in the English data set.

Additional information on the data can be found in the task description paper (Piskorski et al., 2023).

3 System Description

3.1 Experimental Setup

The train, dev, and test data splits were already given. We tokenized them by using a WordPiece tokenizer of a multilingual cased BERT model (Devlin et al., 2018) of `HugggingFace`.[3] To vectorize these tokens we used `tf-idf` (Term Frequency times Inverse Document Frequency) to get features that are independent of the length of the tokenized text. We used a 3-gram range.

An example of a tokenized sentence from article number 111111117 of the English dev set:

```
Original Headline:
Trump formally
nominates Gina Haspel
to be next CIA
director WASHINGTON --


Tokenized Headline: '[CLS]',
 'Trump',
 'formally',
 'nominate',
 '##s',
 'Gina',
 'Has',
 '##pel',
 'to',
 'be',
 'next',
 'CIA',
 'director',
 'WA',
 '##S',
 '##H',
 '##ING',
 '##TO',
 '##N',
 '--',
```

One can see that the tokens are words and subwords. For example, the third person singular marker '-s' gets separated from the word stem 'nominate'. These tokens were used as features for a support Vector machine classifier in the first subtask. For the second subtask, we used the same features in a multi-class logistic regression model. For all experiments used a multilingual BERT model (Devlin et al., 2018) to tokenize the texts. After that, we used an SVM (Chauhan et al., 2019) or logistic regression (Sun et al., 2019) implementation from `scikit-learn` (Pedregosa et al., 2011).[4]

3.2 Hyperparameter Setting

For the parameter tuning of the SVM, we used the grid-search algorithm of `scikit-learn`. Since the shared task evaluation measure was F1 macro, the F1 macro score was used for the rankings.

We tried to improve the performance with multiple experiments, also trying with bigger 3-grams but the best results were obtained by using three-grams. We also searched for the best C values. We searched first in log space and then tried to find better parameters in the linear search around the best parameter that we found before. We used a simple grid search to find the best parameters.

3.3 Additional Attempts

Our first approach was to optimize the baseline model. We tried different classifiers and we could slightly increase the F1 score of the baseline from 0.25 to 0.27 by using a Random Forest Classifier instead of SVM and an n-gram range of 5. We also tried bigger n-gram ranges without improving the results. For Italian, the F1-score was 0.387 by using a Random Forest Classifier instead of SVM and an n-gram range of 10, which is worse than the baseline (F = 0.45). This means we could not improve the linear classifier with the help of other classic classifiers alone in a multi-lingual approach.

4 Results

4.1 Subtask 1

In the first subtask, our F1-score on the test set was between about 0.23 (for Greek) and about 0.57 (for German). Our system was worse than the baseline on the languages that it was trained on, but it was better than the baseline for Spanish and Greek, which were the languages for which no training data was given. In Georgian, the third language for which no training data was given, our system was on baseline level (Table 1).

In the German dataset, our system had the highest F1-score. This may be since the class frequencies in the training and the dev set are similar and most probably it is the same in the test data (Table 2). Whereas on the English dataset in the dev set, the data distribution is completely different.

[3] https://huggingface.co/bert-base-multilingual-cased

[4] https://scikit-learn.org/

This may lead our system to ignore the minority classes and predict the opinion class in each case of the test set, although we balanced the classes in the `scikit-learn` classifier to prevent this problem. Our system just modeled the high-frequency classes.

Our ranking was between the 12th to 20th place in comparison to about 15 to 20 participants. Our best rankings we could get in Spanish and German in twelfth place out of 16 participants.

4.2 Subtask 2

In the second subtask, we found a similar pattern. Our model was better than the baseline in Spanish, Greek, and Georgian, for which no training data was given. In addition, it was better than the baseline in German. In the other languages, our system was very close to the baseline (Table 2).

Our ranking was between the 23rd and 10th place in comparison to about 16 to 23 participants. Our best rankings we could get in Spanish and Georgian in tenth place out of 16 participants.

	SUBTASK 1	
language	FramingFreaks	baseline 1
Spanish	0.32	0.15
Greek	0.23	0.17
Georgian	0.26	0.26
English	0.24	0.29
Italian	0.36	0.39
Russian	0.24	0.40
French	0.34	0.57
German	0.57	0.63
Polish	0.28	0.49

Table 1: Table for subtask 1 with the macro F1-scores of our system in comparison to the baseline for the different languages on the test set. The F1-scores, where our system was better than the baseline are marked in green.

5 Post-analysis of Subtask 1

We run all the following experiments on the test set.

5.1 Monolingual Models

We assumed that our subword models may not generalize well, because the structure and morphology that they should capture are so different in the languages. So we tried to rerun our model with monolingual BERT models to see if the results could be improved in that way.

	SUBTASK 2	
language	FramingFreaks	baseline 2
Spanish	0.22	0.12
Greek	0.38	0.35
Georgian	0.35	0.26
English	0.20	0.35
Italian	0.45	0.49
Russian	0.22	0.23
French	0.33	0.33
German	0.55	0.48
Polish	0.56	0.60

Table 2: Table for subtask 2 with the macro F1-scores of our system in comparison to the baseline for the different languages on the test set. The F1-scores, where our system was better than the baseline are marked in green.

We chose Italian and German because these were the two languages, in which we performed best regarding the F1-score. We used a German cased BERT model of huggingface[5] and an Italian cased BERT model.[6] But with a new F1-score of 0.51 for German and a new F1-score of 0.30, we were worse than before for both languages.

6 Conclusion

The first subtask was difficult because the datasets were not balanced and the input length of the texts differed between small paragraphs and longer news articles. The first fact results in the problem that the majority class of the training set is predicted over proportional often. The different lengths of the texts could lead to a classification that is just based on length. This may be due to the fact, that news articles are longer, in general, and therefore the class 'reporting' is just predicted by the size. The difficulty in subtask 2 was to not just find one or two of the correct framings but all of them.

The majority class in the English dev set was opinion. This lead our system to ignore the minority classes and predict the opinion class in each case of the test set, although we balanced the class weights to prevent this problem. Our system just modeled the high-frequency classes. We assumed that a bag-of-word model should be worse, especially for morphologically complex languages, but the baseline used these bag-of-word models and

[5] https://huggingface.co/bert-base-german-cased

[6] https://huggingface.co/dbmdz/bert-base-italian-cased

was better in most of the languages. A monolingual setup also did not improve our results.

Our analysis of the data shows that our system does not seem to classify meaningful features but takes the frequency of the training data and low features like the article length as hints for categorization. Maybe the use of a transformer model or other language model that takes the relationship of the words into account would have led to a better result. Methods for data argumentation like oversampling could have been helpful as well.

Acknowledgments

Cordial thanks to Çağrı Çöltekin, who made this project possible and helped us a lot.

References

Vinod Kumar Chauhan, Kalpana Dahiya, and Anuj Sharma. 2019. Problem formulations and solvers in linear SVM: a review. *Artificial Intelligence Review*, pages 803–855.

Çağrı Çöltekin and Taraka Rama. 2018. Tübingen-Oslo at SemEval-2018 task 2: SVMs perform better than RNNs in emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 34–38, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding.

Hugo Jair Escalante, Thamar Solorio, and Manuel Montes-y Gómez. 2011. Local histograms of character n-grams for authorship attribution. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 288–298, Portland, Oregon, USA. Association for Computational Linguistics.

Qi Han, Junfei Guo, and Hinrich Schuetze. 2013. CodeX: Combining an SVM classifier and character n-gram language models for sentiment analysis on Twitter text. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 520–524, Atlanta, Georgia, USA. Association for Computational Linguistics.

Georgiana Ifrim, Gökhan Bakir, and Gerhard Weikum. 2008. Fast logistic regression for text categorization with variable-length n-grams. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 354–362.

Artur Kulmizev, Bo Blankers, Johannes Bjerva, Malvina Nissim, Gertjan van Noord, Barbara Plank, and Martijn Wieling. 2017. The power of character n-grams in native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 382–389, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Jakub Piskorski and Guillaume Jacquet. 2020. TF-IDF character N-grams versus word embedding-based models for fine-grained event classification: A preliminary study. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 26–34, Marseille, France. European Language Resources Association (ELRA).

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.

Yuan Sun, Zhihao Zhang, Zan Yang, and Dan Li. 2019. Application of logistic regression with fixed memory step gradient descent method in multi-class classification problem. In *2019 6th International Conference on Systems and Informatics (ICSAI)*, pages 516–521.