Data Set for Stance and Sentiment Analysis from User Comments on Croatian News

Mihaela Bošnjak and Mladen Karan
Faculty of Electrical Engineering and Computing, University of Zagreb
mihaella.bosnjak@protonmail.com, mladen.karan@fer.hr

Abstract

Nowadays it is becoming more important than ever to find new ways of extracting useful information from the evergrowing amount of user-generated data available online. In this paper, we describe the creation of a data set that contains news articles and corresponding comments from Croatian news outlet *24 sata*. Our annotation scheme is specifically tailored for the task of detecting stances and sentiment from user comments as well as assessing if commentator claims are verifiable. Through this data, we hope to get a better understanding of the publics viewpoint on various events. In addition, we also explore the potential of applying supervised machine learning models to automate annotation of more data.

1 Introduction

In the world of unceasing connectedness there is a constant surge of user-generated data online. On news outlets a multitude of opinions and reactions are present. Such amounts of data are too large to analyze manually. On the other hand, automated analysis of this data is difficult due to its inherently unstructured nature. Models that could automatically and efficiently extract structured information from large amounts of data would save time, energy and yield valuable information. We propose a structured annotation scheme that labels claim verifiability, stance, and sentiment on news outlets.

Information about stance, can provide an overview of public opinions and information about currently favorable political movements. Furthermore, claim verifiability can help the fight against fake news, as automated verifiability detection could bring forth claims that are not verifiable and that could potentially be just a rumor or simply made up. Moreover, the data set could be analyzed in search of interactions between the labels. The contribution of this paper is twofold. First, we create a data set of user comments on news in Croatian annotated with claim verifiability, stance, and sentiment. Second, we perform preliminary experiments with several machine learning models on this data set. We present a general overview of the entire data set creation process with caveats and experimental results.

2 Related Work

For stance detection similar definitions of labels can be found in Mohammad et al. (2016) and Zhang et al. (2018). For claim detection we have strongly relied on Park and Cardie (2014) and Guggilla et al. (2016) when building our definitions of claim labels. An overview of approaches and labels for fake news detection can be found in Zhou and Zafarani (2018). For a good general overview of sentiment analysis or opinion mining we refer to Pang and Lee (2008).

3 Data Set

3.1 Data Source

To collect data we have turned to a Croatian news outlet *24 sata* (www.24sata.hr). We chose this outlet for practical reasons, as *24 sata* covers more shocking, diverse, and popular news. Thus, people commented more on this outlet. Most comments on this website contained noisy user-generated text expressing a wide range of stances and sentiment.

The data was scraped from three categories: *newest*, *trending*, and *news*. Articles were scraped and updated on a daily basis and new comments were added to articles old up to one month. We selected news articles for the annotation at random, ignoring those with less than five comments, as we wanted to focus on articles that peeked public interests. Furthermore, from each article, we select a random subset of comments for annotation. Comments that are considered are 'root' comments. These comments could have responses to them but

Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing, pages 50–55,
Florence, Italy, 2 August 2019. ©2019 Association for Computational Linguistics

they themselves are a response only to the article. Simply said they are the first comment in a thread. We didn't want to use comments that were in threads as that would additionally make the annotation process more complicated. Annotators would have to read the whole thread of comments and understand the main topic, arguments and the discussion that is lead. Also, a lot of labels such as *stance* should be revised to take into consideration former comments. We do understand that all comments may have some influence on the commentator and that they could be take into consideration. As we do not know the measure or significance of that influence we are not bringing any more complexity to an already complex process without knowing if we would reap any benefits.

3.2 Annotation Scheme

In a search for an adequate annotation system, we considered the reason people comment on these outlets and what we expected to gain. Most comments were not carefully curated sentences that were there to inform other readers. They were bursts of reactions, insults, compliments, opinions, etc. People commented because they were enticed by the news content enough that they had to express their inner opinions publicly. Some wrote sentences to inform, others to support or judge, but these are all speculations behind users motivation. Because of their spontaneous creation comments varied in size, structure, and purpose. The main question was how to structure something of this complexity without losing important details?

We have tried to answer that question with the following set of labels, motivated by similar schemes from Mohammad et al. (2016); Park and Cardie (2014). There are three main categories called *Claim*, *Stance*, and *Sentiment*. With these three groups, we are deconstructing a comment to three separate parts. There is a total of 8 labels, most of which are mutually non-exclusive. All annotations are made on the comment-level. We have taken into consideration EDU-level annotations. Considering the complexity of the labels and a limited time out annotators could dedicate we have for now opted for a comment-level annotations. Next we describe all label groups in detail.

3.3 Claim Label Group

Within the *Claim* group we wish to determine the type of the comment with respect to claims therein. Namely, whether it contains a claim. And if so, can we verify it? We take interest in claims that can be objectively verified as we try to divide the claim domain mainly into two groups by standards that are appropriate for the given domain. This group contains 4 labels: *Spam*, *Non-Claim*, *Verifiable* and *Non-Verifiable*.

Verifiable – this label is assigned to comments that contain claims that can be objectively verified regardless of the subjective nature through which they are presented. E.g. *"I think the earth is flat."* Even though it is an opinion it can be objectively verified. Also, all quantifiable claims are considered verifiable regardless of the measure through which they are expressed as long as we know the metric under comparison (Park and Cardie, 2014). E.g. *"I had a lot of water." A lot* is subjective but it can be determined how much water you had or even if you had water. The term of degree is only something to be settled.

Non-Verifiable – comments that are labels this way contain claims that can't be verified objectively. Claims that talk about the future(E.g. *"In two moths it will rain"*), are simple sentences that only contain an adjective and are descriptive (E.g. *"That cat is boring"*) or are private facts (E.g. *"I have two sons"*) (Park and Cardie, 2014).

Spam – this label is here for everything that is unrelated to the news. If the news is talking about cheese then a comment about turtles is spam.

Non-claim – this label is added to cover everything that does pertain to the news article but is not a claim, i.e., does not belong in any of the groups above. This group contains mostly questions, imperative sentences and anything that is borderline. E.g., *"These crooks should be put in prison."* and we arent sure where to put it or if it even belongs to one group.

We point out that the concept of claim in the scope of this annotation does not denote exclusively claims in the classical sense as used in the literature (Aharoni et al., 2014), but also opinions as in Rosenthal and McKeown (2012). Moreover, a comment can contain sentences that fit into all categories. To address this we used the following annotation principle. The comment is first annotated as *Verifiable* and/or *Non-Verifiable* based on whether it contains at least one verifiable/non-verifiable claim. This annotation step is multi-label and the same comment can get both labels if it contains multiple claims of different types. If and only if no labels were assigned in the first step then the comment is

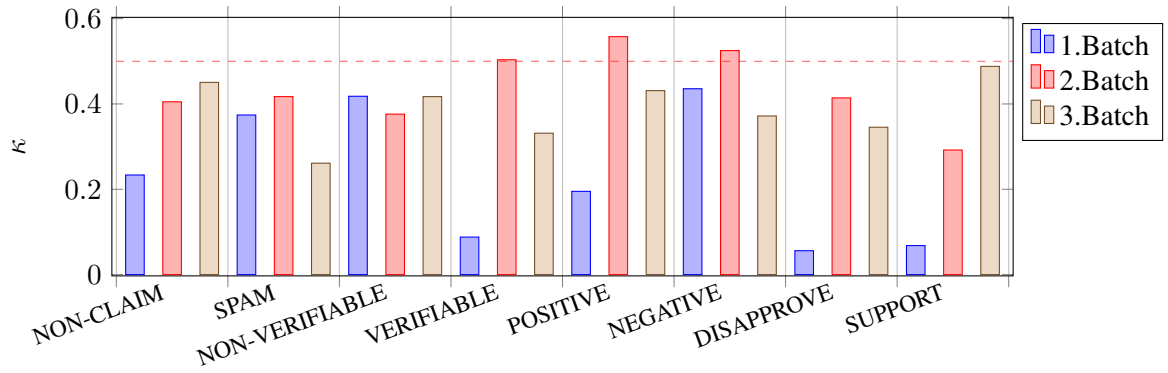

Figure 1: κ over first three batches for each label.

annotated as either only *Non-Claim* or only *Spam*. We acknowledge that some information is lost by this scheme. However, turning this entire group into multi-label would put an additional strain on annotators without much benefit. E.g., from a practical perspective, if a comment contains a sentence that is *Verifiable* it does not provide much additional information to know that it contains another sentence that is *Spam*.

3.4 Stance Group

The *Stance* group contains the *Support* and *Disapprove* labels and is determined *in respect to the title*. We have decided that the title is the target as it would be more difficult to determine stance with respect to the entire article. Also, it would present an additional problem for the annotators since that would make the task more subjective. As the comments on the outlet are not limited by length users often express a multitude of (often conflicting) stances. To allow for multiple stances in the same comment, and to differentiate annotations for comments that are neutral due to several conflicting stances from those truly neutral (with no stance expression), we decided to make this a multi-label task. This contrasts some previous work (Mohammad et al., 2016), where there was a single neutral stance class covering both cases. In our case, a neutral stance is one not containing favorability or interest towards a specific target.

3.5 Sentiment Group

The *Sentiment* group here refers to a manner of speaking. Namely, whether the commentator presents their comment in a positive or negative light. The annotators were instructed to disregard their own sentiment towards the topic of the comment, as this would bias annotations. There are two labels: *Positive* and *Negative*. They are also multi-label for similar reasons as the *Stance* group.

3.6 Annotation Process

Annotators were given written instructions and detailed explanations of labels. Each annotator got an Excel table for each article with comments. For each label, they had to note if that label was present or not while abiding the rules regarding labels explained in the previous sections.

There were 5 annotators in total and 6 batches of data. They annotated independently. On first 3 batches, there were overlaps between all annotators in order to estimate inter-annotator agreement (IAA) and calibrate the annotators. For the first three batches, each batch had two groups and each group annotated one half of the batch. One of the annotators annotated all data of the first 3 batches (was in both groups). After the first three batches, the number of annotators had decreased, so we focused on collecting more data and occasionally checking IAA on some articles to ensure that annotators were still well aligned. In the final data set, we omit the first two batches as the labels have changed a bit during annotation and these batches were meant to calibrate the annotators.

During annotation, we faced two main challenges. First, we could not predict everything that could be in the comments, thus instructions were not perfect in the beginning and had to be revised during annotation. For the same reason, we revised the number of comments sampled per article, as we realized that it was better to take more articles and fewer comments. The revised approach covered a wider range of different topics and thus allowed us to get acquainted with the entire domain faster and made the data set more diverse.

	Claim				Sentiment		Stance	
	Non-Claim	Spam	Verifiable	Non-verifiable	Positive	Negative	Support	Dissaprove
Train	475 (124)	523 (20)	535 (257)	525 (250)	549 (61)	501 (193)	445 (69)	470 (68)
Train(b)	702 (351)	1006 (503)	556 (266)	548 (264)	976 (488)	616 (308)	752 (376)	804 (402)
Dev	205 (42)	162 (6)	172 (71)	215 (104)	131 (21)	148 (70)	222 (25)	224 (24)
Test	224 (42)	219 (6)	197 (91)	164 (75)	224 (21)	255 (88)	237 (32)	190 (30)

Table 1: Splits across labels for training and measuring results. Number of positive examples for each split and each label is in the parenthesis. Train(b) denotes the balanced version of the train set.

Second, because of a complicated annotation structure, it was challenging to calibrate the annotators, especially near the beginning of the annotation when our knowledge of the domain was limited. In the first batch, many annotators did not assign any class to many of the comments. Consequently, we strongly encouraged our annotators to label a comment with something, even if they were not sure of it or found the instructions pertaining to the specific situation unclear. This helped us to better calibrate the annotators as it provided insights into what was unclear and the reasons for disagreement. We did create additional noise with this approach but, we preferred recall over precision as positive examples in our data were generally scarce for most labels. We used Cohen-s κ as an IAA measure. For each label we calculated κ averaged over the annotator pairs as presented in Figure 1. On the graph, we can see the improvement of κ, especially in the stance category. The third batch is slightly worse. The likely cause of this small drop is a slight change in the meaning of labels introduced between the second and third batch. In the final data set, we included the last 4 batches out of 6. For the last 3 batches, we did not calculate total IAA because there were fewer annotators available. However, we did manual checks of agreement for some of the articles and further calibrated annotators through additional detailed explanations.

In total, the data set is comprised of 54 articles and 904 comments with 16.74 comments per article on average. The average lengths (in words) of articles and comments are 330.14 and 25.21, respectively. The least represented class is *spam* with only 32 positive examples. We make it publicly available.[1]

4 Models

There are 5 different models that we tested on this data set. The first is the baseline model which is a linear SVM (Vapnik, 2013). The input of the SVM is the concatenation of TF-IDF weighted vector representations of the news title, news body, and the comment, respectively. We also consider a second SVM model which is similar to the first one, but adds the following features: total word count in the comment (1 feature), and the count and presence in the comment of uppercase letters, question marks, exclamation marks, punctuation marks and negations (10 features total).

We also experiment with some deep learning based models. As the encoder for text we consider convolutional neural networks (Krizhevsky et al., 2012), gated recurrent units (GRU) (Cho et al., 2014), and long short-term memory networks (Hochreiter and Schmidhuber, 1997). We present the text to the encoder as a sequence of word2vec (Mikolov et al., 2013) word embeddings from a word2vec model trained on the HrWaC (Ljubešić and Erjavec, 2011; Šnajder et al., 2013) corpus. We have a separate encoder for (1) the concatenation of the article title and body and (2) for the comment. The outputs of both encoders are concatenated and passed through a linear classification layer. For regularization we perform early-stopping on the dev set. Hyperparameters for these models we considered are given in Table 3 and were also optimized on the dev set. As these are preliminary experiments, we did not perform exhaustive hyperparameter search for the deep learning models on all labels, but only for the more frequent ones, and reused those hyperparameter values for the models dealing with the rest of the labels. Admittedly, deep learning models could possibly yield better performance with more thorough hyperparameter tuning. We used the Adam (Kingma and Ba, 2015) algorithm with minibatch size 16 to train the models.

5 Experiments

For each label, we split the data into a train, dev, and test portions. The splits are disjunctive with respect to the articles, meaning that comments cor-

[1] http://takelab.fer.hr/crocomm/

	Claim				Sentiment		Stance	
	Non-Claim	Spam	Non-Verifiable	Verifiable	Positive	Negative	Support	Dissaprove
SVM	0.351	0.048	0.577	0.547	**0.194**	**0.519**	0.240	0.275
SVM + features	**0.367**	0.053	*0.627*	*0.678*	0.178	*0.471*	0.221	**0.296**
LSTM	*0.254*	*0.235*	0.591	*0.675*	0.167	*0.447*	**0.255**	0.247
GRU	0.337	***0.261***	0.577	0.553	0.152	*0.479*	0.194	0.290
CNN	0.300	0.000	***0.649***	***0.683***	0.154	0.515	0.251	0.231

Table 2: Results of classifiers across all labels. The best result for each label is given in bold. Entries in italic represent results that are statistically significantly better than the SVM baseline from the first row.

Model	Hyperparameter	Values
CNN	Number of kernels	**5**,10,25
	Kernel size	1,**3**,5
LSTM/GRU	Hidden/cell size	10,**25**,50
	Bidirectional	**Yes**, No

Table 3: Hyperparameters considered for the deep learning-based models. The values that were best performing in most experiments are given in bold.

responding to the same article are all in the same split. Furthermore, as the data set is highly imbalanced, we perform the splits in a stratified manner, ensuring the ratio of positive and negative examples is roughly equal for train, dev, and test. Through this, we have ensured that all of our splits contain positive examples. However, an imbalance that can hurt model performance was still present in the train data. To alleviate this issue we artificially balanced the train set by oversampling positive examples until the number of positive and negative examples was equal. This was done for all labels as positive examples were always the minority. For different labels, we had different splits. However, for each label, the same (artificially balanced) train, dev, and test sets were used for all models. In Table 1 we can see the split through the labels. For train we have counted in artificially examples thus the sum through columns isn't the same. We train all models on the train set, optimize hyperparameters on the dev set and report results on the test set.

Some preliminary results are given in Table 2 as F1 score for each label along with statistical significance tests (we used a permutation test on test set predictions). Performance on most labels is rather low, indicating the task is highly complex.

In most cases, adding features to the baseline model improved performance. For labels *Verifiable*, *Non-Verifiable* the differences are statistically significant. On the other hand, on the *Negative* label the SVM baseline is the overall best model. The deep learning approaches were not expected to be very good, as the data set is small, but they do provide some respectable results, mostly for the classes from the *Claim* group.

6 Conclusion

In this paper, we presented a data set for Croatian news annotated with (1) claim verifiability, (2) sentiment, and (3) stance We have managed to calibrate annotators and achieved moderate Cohen κ agreement on this highly challenging task. We also present preliminary results of machine learning based prediction models.

A clear limitation of this work is the small size of the data set. Thus, we envision that in the future much more data could be annotated using the same methodology. This would enable a more meaningful analysis of user behavior and might reveal unobserved connections between labels. E.g., a comment with many claims may be more likely to also express a stance. In a related vein, transfer learning could be applied to such data, in order to exploit such relations between labels by jointly training the models. Another possibility for improving models is including information from other comments in the same thread as well as additional meta-data, Finally, the annotation scheme could be improved by annotating at the level of sentences, which would allow for even deeper further analysis.

Acknowledgments

We acknowledge Jan Šnajder nad Filip Boltužić for fruitful discussions and their input on this paper.

References

Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the First Workshop on Argumentation Mining*, pages

64–68, Baltimore, Maryland. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111.

Chinnappa Guggilla, Tristan Miller, and Iryna Gurevych. 2016. CNN- and LSTM-based claim classification in online user comments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2740–2751, Osaka, Japan. The COLING 2016 Organizing Committee.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. pages 1–13.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Nikola Ljubešić and Tomaž Erjavec. 2011. hrWaC and slWaC: Compiling web corpora for croatian and slovene. In *International Conference on Text, Speech and Dialogue*, pages 395–402. Springer.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(12):1–135.

Joonsuk Park and Claire Cardie. 2014. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland. Association for Computational Linguistics.

S. Rosenthal and K. McKeown. 2012. Detecting opinionated claims in online discussions. In *2012 IEEE Sixth International Conference on Semantic Computing*, pages 30–37.

Jan Šnajder, Sebastian Padó, and Željko Agić. 2013. Building and evaluating a distributional memory for croatian. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 784–789.

Vladimir Vapnik. 2013. *The nature of statistical learning theory*. Springer science & business media.

Qiang Zhang, Emine Yilmaz, and Shangsong Liang. 2018. Ranking-based method for news stance detection. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 41–42, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Xinyi Zhou and Reza Zafarani. 2018. Fake news: A survey of research, detection methods, and opportunities. *CoRR*, abs/1812.00315.