Event Detection using Natural Language Processing Models

Jakub Szumny

Information Science + Data Science | University of Illinois at Urbana-Champaign

jszum3@illinois.edu

ABSTRACT

Nowadays, many serious events are talked about on social media platforms, even more so than on better platforms. Currently, detecting these events that are occurring takes far different approaches, such as waiting for police reports, seeing it firsthand, and other non-efficient approaches. This leads to these events flying under the radar for a good amount of time, allowing them to spiral out of control, leading to many negative consequences. This research is a continuation of the work done by Dong Wang and his team, and their zero and few-shot event detection via prompt-based meta-learning study[1]. In their work, they developed a custom NLP model, to detect events through data. I tested their models on data provided to me by Lanyu Shang, trying to find the best model to work with in the future. The models I tested were the FewEvent Zero-Shot and Few-Shot models, and the Maven Few-Shot and Zero-Shot models. When all of these models were trained and tested

on the dataset I was provided with, the Maven model on Zero-Shot performed the best out of all.

INTRODUCTION

There are so many events/things that take place in the world, that go unnoticed for a very long amount of time. For example, when COVID-19 first started, someone got sick and didn't even know they had the first case of COVID-19, as it wasn't a thing yet. Then once the information spread the first place it spread was Twitter. People were talking about it for a very long time before action was taken to prevent the spread of it. If there was a better way to detect these events from social media, it would be much easier to detect and prevent the spiraling of these events.

Natural language processing models are the best way of predicting events from social media and could make a serious change in the way we predict and find these events in the future. The goal of this research is to test some premade models on a new set of data and to see how well they perform on this data, to get a good starting point for future research.

I have never worked with NLP models before, so this was a very new and difficult process for me, as working with these models and data was completely new to me. Also, I initially was supposed to be working on the project of predicting the biodegradability of plastics by enzymes using machine learning, but it wasn't ready for me at first, so I did a lot of random work before starting this new research. I first was tasked with doing a literature review on data sparsity, and I found many related works on data sparsity issues using transfer learning, contrastive learning, few/one/zero-shot learning, and domain adaptation. I was then tasked with doing a literature review on misinformation and found a few related works involving image and text misinformation, and also video and audio misinformation. Then I was tasked with helping out on a different research project, and manually labeled large amounts of data for misinformation reasons, on Polio and Monkeypox tweets.

After all of that work, I then began this research, by doing a literature review on event prediction on image and text, and also on video and audio. I never worked with NLP models, or anything like this before, so the entire thing was a big learning curve. I had to do this research while also learning how to do this research, so I learned a lot during this semester. Once I started to get the hang of it and tried to replicate the models from the MetaEvent Github[2], I started to face my first challenges. I ran into many issues with the code and had to do a lot of debugging before I could get it to even run with the data that they used. Once I got that code to work, I ran into another problem involving memory, as the datasets used were very large, too much for Google Colab to handle, so I had to deal with changing the batch sizes, editing the model to be able to handle the data better, and also taking subsets of the datasets, so it could run on Colab. Once I got the models to work on the data from the original study and dataset, I was then able to try and run it on the drought data I was given by Lanyu Shang.

I also have never preprocessed data in this way before, so making sure the new dataset was compatible with the model was a very difficult task for me, but I managed to get it to work and get some initial results. The results were not the best, and I believe this to be because I had to train much smaller batches, and also was only able to use a subset of the data, but if these models were trained on a better GPU, then the results would be much better.

DATA

For this research, I used drought data from tweets, which contain thousands of tweets about a drought. I was provided with this dataset by Lanyu Shang, so I am not sure about the exact specifics of the dataset and where it came from. The data from the MetaEvent Github[2] had good preprocessing models, I just had to tweak them to work well with this new dataset, but once I had figured that out, I didn't have any major issues training the models with this data.

METHODS

Subset

As I have mentioned before, the dataset was much too large to be able to train a model on, so I had to take a subset of the dataset to research it. I decided to take only about 25,000 instances from the dataset, which was a good number to be able to get initial results from the models.

Train test split

Now that I had this subset of the drought dataset, I was able to split it into a training dataset, a test dataset, and a dev dataset. I did this using the train_test_split function from sklearn, splitting it at about 70% train, 15% dev, and 15% test. Once I had this all split up, I saved the JSON files and then was able to use them in training the models.

Training the models

I trained each model on only 4 training iterations, as once again the dataset was too large for Colab to handle, and had to lower the batch size to only 5. These are not the most ideal factors in training a model, but I had no other option as free Google Colab only has a decent GPU. I then used the same optimizer as the previous work, the same learning rate, and also the same scaling.

RESULTS

Model	F1	AMI	FM	Dataset
FewEvent Zeroshot	0.1893	0.1921	0.231	Drought Data
FewEvent FewShot	0.2311	0.2287	0.192	Drought Data
Maven FewShot	0.2403	0.1864	0.2003	Drought Data
Maven ZeroShot	0.2732	0.2553	0.2643	Drought Data

FIG. 1. All results of different models and their experiments on drought data

FewEvent FewShot

I first trained the Few Event Few Shot model on the drought data and received an F1 score of .2311 on the testing dataset. This is a substantial decrease from the results the MetaEvent team[1] received, but I believe it would be on par with the results if I had a better GPU available.

FewEvent ZeroShot

After training the few event zero shot model on the drought data, I received a lower score than that of the few shot model, with a 0.1893 F1 score. I believe this to be accurate as the score was also lower in the MetaEvent study[1] on zero-shot than on few-shot, for either the few event or maven.

Maven FewShot

After testing the Maven few-shot model on the drought data, I received an F1 score of about .2403. It performed slightly better than the FewEvent, but still nowhere near as good as the MetaEvent study[1].

Maven ZeroShot

I received a .2732 F1 score on testing the Maven zero-shot model, which was my best result on this data set out of any other model. Again, I was only able to train under non-ideal training circumstances, so I believe that if I had a stronger GPU the results would be much better, on par with that of the previous study[1].

DISCUSSION

Impact

I believe that this work is a great starting point for the future of event detection. Even though I did not get the best results, I was able to test the models from the MetaEvent study[1], on a new dataset, and to see how well it performs on that new data. I have also created a Google Sheets of a literature review, on many different studies on this topic as well, which can be used for future research as well.

Comparison to past works

Compared to the past work of MetaEvent[1], my results were very subpar to that of their results. I did of course have to train the models in non-ideal ways, and so I believe if the models were trained on the same data, but with a stronger GPU and ideal ways, it would provide results just as strong as that of MetaEvent.

Future Directions

Now that I have started the training and testing of these MetaEvent models[1] under new data, the first next step would be to train these models on the same data, but in a better environment with a stronger GPU, to be able to fully train the models effectively and efficiently, to produce greater results. I have placed all of my work in a folder shared with Lanyu Shang, and hopefully, that work can be used for future work.

CONCLUSION

This research provides a starting point for event detection using Twitter data and provides initial results on the effectiveness of these models from the MetaEvent study[1] by Dong Wang and peers. I have also personally learned a great deal of information on Natural Language Processing models, and how to work with them, which I didn't know prior. I am very glad to have been able to conduct this research, as it helped me learn a lot, and also furthered my knowledge on this topic, which I can use in the future. I hope that this work can be used to better event detection in the future.

ACKNOWLEDGMENTS

I conducted this work under the guidance of Lanyu Shang, under the broader project by Dong Wang. Thank you very much for this opportunity!

REFERENCES

1. Yue, Z., Zeng, H., Lan, M., Ji, H., & Wang, D. (2023). Zero- and few-shot event detection via prompt-based Meta Learning. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. https://doi.org/10.18653/v1/2023.acl-long.440

2. Yueeeeeeee. (n.d.). *Yueeeeeeee/Metaevent: [ACL'23] pytorch implementation of zero-and few-shot event detection via prompt-based Meta Learning*. GitHub. https://github.com/Yueeeeeeee/MetaEvent