CRF for Target Extraction of COVID-19 Events on Twitter

Xinyi Su
McCourt School of Public Policy
Georgetown University
xs132@georgetown.edu

Abstract

In this paper, I use a CRF model to identify relevant COVID-19 event keywords from tweets. The data is gathered using Twitter API with the help of the script from W-NUT 2020 Shared Task #3 (http://noisy-text.github.io/2020/extract_covid19_event-shared_task.html), for the two COVID-19 events "test positive" and "death". The goal of this study is to apply the CRF model to figure out whether the keywords for the relevant COVID-19 event indicate a real COVID case in life, which is similar to identifying self-reported COVID-19 cases from Twitter. To simplify the task, I tag the keywords for each event and convert target extraction into a classification task. The features added to the model include word type, word POS, word shape, and the word itself. As for the result, the second CRF model increases the accuracy scores by 10 percentage points from the baseline model.

1 Introduction

Nowadays, as more and more people are sharing their life on social media platforms, social media platforms like Twitter and Facebook can thereby provide up-to-date information of current events from the public and serve as a channel for policymakers and businesses to get access to public opinion what is actually happening in people's life.

For the reasons above, growing attention has been paid to social media data analysis, and information extraction is one of the most popular tasks. In this paper, I will focus on the public opinions regarding the COVID-19 pandemic on Twitter and try to find an approach to distinguish the tweets representing real COVID-19 cases from those only covering some general comments regarding the pandemic. In this sense, this task is essentially the same as automatically identifying "self-reported" (or people-reported) COVID-19 cases and the real idea behind each tweet will be extracted and examined.

The Conditional Random Field (CRF) model is commonly adopted to recognize trends and patterns. It will take the features, sequence and the context of the target expressions into account. To accomplish the aforementioned task, I use customized word tags to mark specific COVID-19 event keywords prevalent in the whole corpus, based on the original labels from the annotated dataset provided by W-NUT 2020 Shared Task that can tell if this tweet represents a real case.

Meanwhile, as Sristy, N. B. et al. (2017) summarize in their paper, social media data has "unusual characteristics like short length, stylistic variations, acronyms, noisy and unstructured forms", which will bring sufficiently more challenges in data cleaning and evaluation. Tools and data pre-processing methods have been developed to solve the problem, and in this paper, I will use a developed package called "tweet-preprocessor" in combination with word feature tags to alleviate the negative impacts of unstructured noisy data.

2 Related Work

Information extraction in natural language processing is a task to automatically extract target information from unstructured data sources (Wikipedia). In the past decade, researchers in this field switch their attention from news articles to social media data which can serve as a good complementary source of information but much noisier and more disorganized for automatic extraction. Allen et al. use latent variable models

and developed the first open-domain event extraction and categorization tool TWICAL particularly for Twitter data (Ritter et al., 2012). In addition to open-domain approaches for event extraction, there are also domain-specific approaches that are adopted when the keywords or event types are known in advance, and they are usually based on supervised learning methods (Xu et al., 2019). And the task of this paper is derived from the idea of domain-specific Event Extraction, though they are not exactly the same.

CRF has been applied in information extraction for a while. Hamdan et al. (2015) extract Opinion Target Expressions (OTEs) in restaurant reviews with a CRF model and further apply a Logistic Regression model to conduct aspect-based sentiment analysis. They include various features, including lexical, syntactic, semantic and some other features, into their feature function and also add the features of 2 or 3 previous and subsequent objects into their list, even though the CRF classifier has already taken context into account.

As for relevant studies dealing with Twitter data, in the research of Sristy, N. B. et al. (2017) to extract events in Indian language Twitter data, they intentionally use POS tags in their CRF model to address the problem of out-of-vocabulary in social media data. Xu et al. (2019) employ a CRF layer in their task to "Jointly Detect and Extract Events from Twitter". This study points out the fact that event extraction labels have strong dependencies among each other, and CRF can help capture the relationships in the neighborhood. Though my task for this paper is not a full Event Extraction task, it's still valuable to consider the context information in order to identify the "self-reported" case and classify the keywords.

In the context of the COVID-19 pandemic, Zong et al. (2020) collected a corpus of 7,500 tweets annotated with COVID-19 events, including "test positive", "test negative", "cannot test", "cure and prevention", and "death". In their baseline model of automatic event extraction, they apply a Logistic Regression model and a BERT model respectively to predict if the candidate slot of the tweet answers its designated event specific question. The results show that BERT model has significantly better performance as it has "a deeper sense of language context" (Horev, 2018). In this paper, I compile my data from the same database for two out of the five events listed above and conduct a relatively different task.

3 Experiments

3.1 COVID-19 Twitter Event Corpus

The original dataset is gathered with Twitter API and the script from the W-NUT 2020 Shared Tasks #3, which can combine the original tweets with the annotations. The script is available at: https://github.com/viczong/extract_COVID19_events_from_Twitter/blob/master/load_data.py. Instead of 7,500 mentioned in the description of the shared task, following the aforementioned method, the number of tweets I eventually collected is only 6,326. It is possible that some of the tweets are no longer available on Twitter after the shared task was published. For this project, I will experiment with the CRF model on the data for two out of the five COVID-19 event types, namely "test positive" and "death". The "test positive" corpus has 2,168 tweets, and the "death" corpus has 1,073 tweets.

This corpus collected coronavirus related tweets from 2020/01/15 and 2020/04/26, annotated by crowd workers on Amazon Mechanical Turk. There are two levels of labels, and I only use the first level which indicates if this tweet captures any real-world "self-reported" COVID-19 cases. This information is stored in the binary label *[response]* for each line of tweets.

To pre-process the tweets, I rely on the preprocessor tool `tweet-processor` developed by Said Özcan. This tool can help remove URLs, hashtags, mentions, emojis in the tweets. Meanwhile, I also lemmatize and tokenize the words in each tweet to make sure that I can mark the keywords precisely.

To create the labels, I tag the keywords *"test", "positive", and "result"* for the event type "test positive", and the word *"die", "dead", "death", "pass", and "away"* for the event type "death". For the keywords in the tweet representing a real COVID-19 case, I tag them by the combination of their event type and response type. All the other tokens are tagged as "Irr" in short for Irrelevant. Two examples for this tagging scheme is shown below:

Tweet#1: *"lol, my friend has started to cancel classes cos a first-year gets tested positive for corona…"* (Crowd workers mark this tweet as "Yes" in response to the question "if this tweet captures a real "self-report" COVID-19 cases")

Tags: *"Irr Irr Irr Irr Irr Irr Irr Irr Irr Irr Irr Irr P-Yes P-Yes Irr Irr Irr"*

Tweet#2: *"Doctors are faking medical charts to say the patients died due to something else other than corona…"* (Crowd workers mark this tweet as "No" in response to the question "if this tweet captures a real "self-report" COVID-19 cases")

Tags: *"Irr Irr Irr Irr Irr Irr Irr Irr Irr D-No Irr Irr Irr Irr Irr Irr Irr Irr"*

As every single token in the corpus is annotated, I split the train and dev data by the ratio of 0.3. Please find the characteristics of the dataset below:

Table 1: Dataset Characteristics

Data	**Number of Tweets**	**Event Keyword Tags**	**Number of Event Keyword Tags**
Train	2268	P-Yes	2432
		P-No	1208
		D-Yes	448
		D-No	666
Dev	973	P-Yes	999
		P-No	421
		D-Yes	235
		D-No	317

3.2 CRF Model and Features

In preparation for training the CRF model, I collect for each token in a sentence the following features for the token itself and also the 2 previous and subsequent tokens (the neighbors).

-Word shape (capital letter).

-Word type (digit, punctuation)

-Keyword (mark "P" for "test positive" keywords and "D" for "death" keywords).

-Word POS using NLTK parser.

-Word lemma from NLTK.

-Stop words from NLTK English.

For the second set of feature list, based on what I have for the first feature set, I add the features of word POS and word lemmas of the neighbor tokens into the list.

Then, the CRF model can be employed as each token in the corpus is annotated with an event keyword tag. I use the `CRFSuite` from NLTK to train the model and make predictions.

3.3 Results

To evaluate the performance, I remove all the tokens annotated with the "Irr" tags. They are useful in training the model but cannot reflect the true performance of the model, since I have already added an event type feature that can tell D for "death" events and P for "test positive" events in the model, and it would be very hard for tokens to be mistakenly marked as "Irr".

Given that the sample data is not perfectly balanced, especially in the "test positive" corpus, for the baseline model, I assign all the "P" and "D" tags to the most frequent class in the corresponding corpus. For example, as shown in Table 1, "P-Yes" is more commonly seen than "P-No", then I will predict all "P" tags to be "P-Yes" in my baseline model. The accuracy scores of this simple baseline model are shown in Table 2.

Table 2: Performance of Baseline Model

Corpus	Test Positive	Death	Full Corpus
Accuracy	0.667	0.704	0.574

The performance for the CRF model is measure with the evaluation matrics: accuracy, precision, recall, and F1 scores. All results are summarized in Table 3.

Table 3: Performance of CRF Models

Corpus	Test Positive	Test Positive	Death	Death
Model Name	CRF (base)	CRF (add neighbor features)	CRF (base)	CRF (add neighbor features)
Accuracy (corpus)	0.709	0.768	0.709	0.768
Accuracy (event)	0.73	0.656	0.777	0.746
Precision	0.755	0.775	0.802	0.788
Recall	0.916	0.294	0.909	0.553
F1	0.828	0.426	0.852	0.65

Compared with the accuracy scores for the baseline model, the CRF models significantly outperform no matter in the whole corpus or the corpus of every COVID-19 event. Particularly, the overall performance increases by 10 percentage points with the second CRF model. The results confirm that CRF models have shown at least some values in identifying "self-reported" COVID-19 cases, and the merits can possibly come from the strength of CRF in processing short and

unstructured text with multiple features, such as word POS, and taking the context information into account. The first strength can be utilized in noisy data processing, while the second one is helpful in Information Extraction, especially Event Extraction which emphasizes the relationship between elements (such as who, when, where) detected in text. Though my task in this paper is not exactly a full Event Extraction task, it also values the importance of finding the correlations among the context information.

As shown in Table 3, the second CRF model using the feature list that includes (+-2) neighbor features clearly has better performance in almost all evaluation metrices. Again, this finding is consistent with previous studies of Hamdan et al. (2015) and shows that the features of the neighbors are useful in CRF models for Event Extraction.

However, given the performance of the baseline model which has the simplest ever form. The first CRF model that doesn't include the neighbor features doesn't do a very good job, as it only increases the overall accuracy by 3 percentage points. One possible explanation behind this, except for the model selection, is that I may fail to fully clean and pre-process the Twitter data. Also, the feature list may not be able to capture all useful information thoroughly. For example, maybe Named Entity Recognition IOB tags should be added into the model, because NER tags can also provide a wealth of predefined category information for each word in the tweets.

Table 4: Confusion Matrix for the CRF Model #2

	Predictions				
Prediction	D-No	D-Yes	Irr	P-No	P-Yes
Label					
D-No	282	35	0	0	0
D-Yes	103	130	2	0	0
P-No	0	0	2	195	224
P-Yes	0	0	0	91	908

Note: This CRF model uses the feature list that covers (+-2) neighbor features.

Both Table 2 and 3 show that the models work better on the "test positive" corpus than on the "death" corpus. The difference can possibly be attributed to the dataset characteristics. For instance, the "death" corpus has substantially fewer data points. However, it's not even close to assuming that the differences may come from the natures of two different events.

4 Discussion

Future work can be done to experiment with other possible solutions for this classification task. For example, a Logistic Regression model can be used either to predict the tags for keywords, which is exactly what I do in this paper, or to predict the labels for each line of tweets, using Bag-Of-Words features or n-gram features. Both approaches can work to identify "self-reported" COVID-19 cases. Similar to Logistic Regression, Support Vector Machine, Naïve Bayes, and Random Forest models can follow this approach as well.

In addition to the classifiers listed above, future work can also experiment with word embedding techniques for the task of Information Extraction, including Event Extraction. Word Embedding can identify word semantics by tracking the word distribution. Ertugrul et al. (2017) use Word Embedding on the Twitter data in the Turkish language, and they adopt a hierarchical clustering method to group the vectors of word embedding. Similarly, Sun et al. (2019) proposed a new Event Extraction model based on Word Embedding that can control the negative impacts of potential spam in tweets and generate an event summary at the same time, in addition to detecting keywords of events.

Acknowledgments

Many thanks to my instructor of ANLY521 Computational Linguistics Adv Python, Prof. Liz, for providing a wealth of knowledge in her wonderful lectures and labs. I couldn't have accomplished this without her supports.
I would also like to thank the new friends from Data Science Department I made in this class. We shared a lot of happy moments during the time we work together, and I even got a perfect second-hand digital piano that I have been looking for for a long time from one of my classmates in this course!

References

Sristy, N. B., & Krishna, N. S. & Somayajulu, D. V. L. N. (2017). Event extraction from Social Media text using Conditional Random Fields. Forum for Information Retrieval Evaluation. http://ceur-ws.org/Vol-2036/T5-3.pdf

Hamdan, H., Bellot, P., & Bechet, F. (2015). Lsislif: CRF and Logistic Regression for Opinion Target

Extraction and Sentiment Polarity Analysis. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 753–758. https://doi.org/10.18653/v1/S15-2128

Ritter, A., Mausam, Etzioni, O., & Clark, S. (2012). Open domain event extraction from twitter. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '12, 1104. https://doi.org/10.1145/2339530.2339704

Zong, S., Baheti, A., Xu, W., & Ritter, A. (2020). Extracting COVID-19 Events from Twitter. ArXiv:2006.02567[Cs].http://arxiv.org/abs/2006.02567

Horev, R. (2018, November 17). *BERT Explained: State of the art language model for NLP*. Medium. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

Xu, M., Zhang, X., & Guo, L. (2019). Jointly Detecting and Extracting Social Events From Twitter Using Gated BiLSTM-CRF. *IEEE Access*, *7*,148462–148471. https://doi.org/10.1109/ACCESS.2019.2947027

Jianqiang, Z., & Xiaolin, G. (2017). Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis. IEEE Access, 5, 2870–2879. https://doi.org/10.1109/ACCESS.2017.2672677

Ertugrul, A. M., Velioglu, B., & KARAGOZ, P. (2017). Word Embedding Based Event Detection on Social Media. 3–14. https://doi.org/10.1007/978-3-319-59650-1_1

Sun, X., Liu, L., Ayorinde, A., & Panneerselvam, J. (2021). ED-SWE: Event detection based on scoring and word embedding in online social networks for the internet of people. Digital Communications and Networks. https://doi.org/10.1016/j.dcan.2021.03.006