THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

INFS 7203 Data Mining

Event Forecast

Occurrence Forecast and Hot Event Prediction using Deep Learning

Yi Liu: 44028156
Yang Li: 44560544
Kexuan Xin: 44021625
Shijie Zhang: 43777644

1. Background and Problem Definition

1.1 Background

Nowadays, fast development of hardware performance and information technology has largely aroused the public passion of data mining upon big data. And recently, event forecasting based on a large collection of daily event data is becoming an increasingly heating topic in this information-explosion age because it is potentially profitable and valuable to make feasible decision for government or enterprises.

The event forecasting in this project generally involves two tasks, one is event occurrence prediction, which is to forecast the trend of what will happen in the future. The other one is hot events prediction by predicting how popular or how hot the events will be.

Time series prediction is one of the most vital domain of this event prediction where observations of the same variable are collected and analyzed to develop a model describing the underlying relationship [1]. Next, the model is utilized to extrapolate the time series into the future, which is especially useful when there is almost no valuable information available from the underlying data generating process or when no proper explanatory model which is relevant to the prediction variable to other explanatory variables. Plenty of models for time series prediction have been developed. Fazle Karim [2] et.al. proposed a model consisting of CNN (convolutional neural network) and LSTM (long short term memory) to predict the price of products using UCR datasets, and achieved state-of-the-art results. Another example, G.P. Zhang et.al. indicated that autoregressive integrated moving average (ARIMA) model is one robust and generally applied models, because its statistical properties and good Box-Jenkins methodology [1].

Hot event prediction is related to text analysis. It involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition and so on. The most important aim for text analysis is to transform text into data for analysis through natural language processing (NLP) and other analytical methods [3].

1.2 Problem Definition

This project aims to perform the event forecasting. It has two main tasks, one is to predict the occurrence of different types of events in the future, using time series prediction model. This provides valuable reference information for governments, enterprises and travelers, etc. The other task is hot news recommendation using text analysis model, which is to forecast if some events in news will be popular by predicting the number of mention times. This may help recommend useful information and provide more friendly user experience when browsing news pages.

In this project, we use GDELT 1.0 Event Database from 4/2013 to 10/2017. This database is a collection of events extracted from news on the Internet. We focus on 3 key attributes, which are event code, NumMentions and Source link. Event code is used to represent event type and it is the event prediction results. NumMentions is the number of mentions of a specific event, which is the result of heat rate of news. Source link is the link of the source news, which are used to acquire the content of news.

2. Preprocessing

2.1 Remove useless attributes and count frequencies

Since there are more than 30 attributes provided in the dataset, choosing the meaningful attributes, such as event code, action geolocation, event date, source link and so on, can help

reduce the size of our dataset and process the data more efficiently (The dataset is decreased from 100GB to 26GB by this approach). Then, the cleaned dataset is input into the MySQL database for counting the frequency grouping by different locations or datetime or event types. The counted result is used for future clustering and event occurrence prediction.

2.2 Crawl news text and match with number of mentions

A web crawler is created to get the original news content based on the source links provided by the dataset. Nearly 1,400 pieces of news is crawled from 12 main news media websites such as www.abc.net.au, www.news.com.au, www.theguardian.com, although some old news in 2013 is deleted by news sites, which may decrease the text analysis model's accuracy in practice. The crawled data are matched with the number of mentions attribute in the dataset according to the same event id. And this news data will be used for the text mining in the future.

2.3 Preprocessing for text analysis

To make the crawled news content appropriate for text mining, a set of text preprocessing steps are needed. Firstly, the stop words and common words are removed, because these words are meaningless but take a large percentage of the total frequency of words in the content (e.g. two most common words "the" and "of" make up about 10% of all word occurrences in text documents). As a result, this may make the dataset noisy and affects the final analysis accuracy. Secondly, all the words are converted to the most common stems by algorithmic-dictionary stemming methods to cut down the size of vocabulary, though this may cause a bit information loss.

3. Methodology and Results

3.1 Event occurrence prediction

Time series prediction model is adopted in this task. As mentioned in the section one, most existing times series predictions focus on predicting a singular value or event on a certain day, such as stock index. In this part, we are trying to predict the occurrences of all 20 base types of events together, and this could be more reasonable because different events may influence each other.

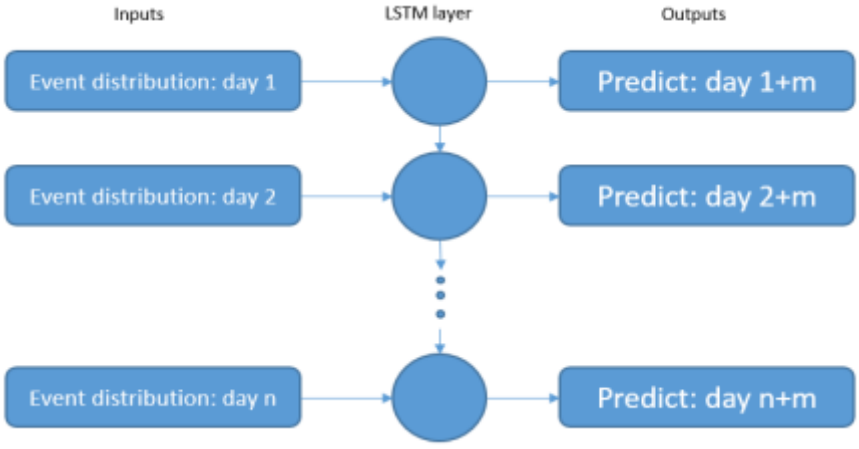

Figure 1 Time series prediction model

Training and prediction on each selected country will be implemented separately using same model. For example, when predicting country A, only the preprocessing results of A will be used for training, then the trained model will only be used for predictions of A. Intuitively, a model trained on A might not be suitable for prediction for B, because different countries have different natures, situations, conditions, etc.

3.1.1 LSTM model for time series

As shown in figure 1, each input is a 20-dimensional distribution vector, in which each element is the frequency of a corresponding event type. The number of inputs is the length of history to consider. Similarly, the outputs are the predicted distribution vector, and "m" is an adjustable parameter indicating which day in the future to predict.

LSTM layer is used as the hidden layer in this model. Basically, LSTM is a recurrent neural network (RNN) variant, which get over some computation problems (vanishing problem) of standard RNN. And RNN is a widely used deep learning structure. The reason why we select RNN structure is that RNN neurons can receive the processing results from other neurons in the same layer, as shown by the vertical arrows in figure 1. For example, when the bottom neuron in figure 1 predicts the vector on day n+m, it receives the vector on day n, as well as the data from top neurons, and this data might contain processed information about previous days (day 1, 2, 3, …) and predicted information on day 1+m, 2+m, … Obviously, this in-layer connection can improve the prediction performance significantly. We use Python language and Sequential, LSTM methods of Keras library to build the LSTM layer.

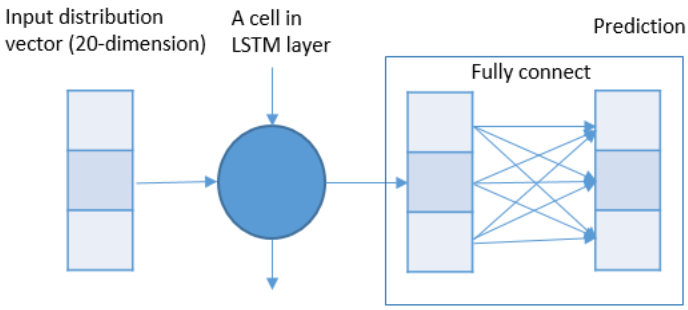

Figure 2 A single segment in the RNN chain

Unlike some LSTM model used for predicting singular value, where only one fully connected layer is used, in this model, each LSTM cell should be followed by a separated fully connected layer, so that the output is a series of 20-dimensional vectors. The figure 2 shows the detail structure of one single segment in the RNN chain. This structure can be achieved by using TimeDistribute and Dense methods of Keras library.

3.1.2 Test results and analysis

As for the evaluation method, the mean absolute error (MAE) is commonly used for regressions and numeric predictions. Assume the true values and predicted values are x, and y, $MAE = \frac{1}{n}\sum_{i=1}^{n}|y[i]-x[i]|$. To reflect the error more straightforward, we use normalized MAE (NMAE), which is $\frac{1}{n}\sum_{i=1}^{n}|y[i]-x[i]|\,/\,x[i]$, so that the error can be normalized no matter the actual values of x are small or large. Currently, if the parameter m is 10 (predict 10 day's future)[1], the best NMAE is 0.11, and obtained when predicting Australian using 30 day's history. The line diagram below shows prediction results in China compared to the truth.

[1] It might be useless if m is too small

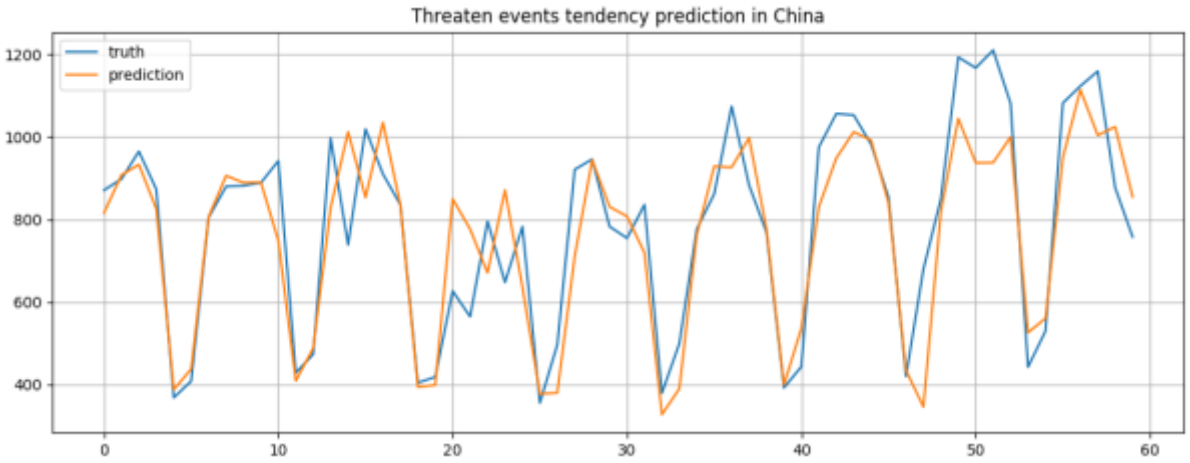

Figure 3 Threaten events tendency prediction in China

The following table shows the results for different countries and different "m". Basically, the larger the "m", the higher the error.

Country	Australia	UK	China	India	US	Syria
NMAE	0.11	0.12	0.22	0.17	0.20	0.41

Train and predict for selected countries (consider 30 day's history, m = 10)

m	10	15	20	25	30
NMAE	0.11	0.14	0.26	0.29	0.34

Train and predict using different "m" (use data of Australia, consider 30 day's history)

In the experiments, except the model illustrated above, we also tested some other model structure, such as adding dropout layer after LSTM layer, or using bidirectional LSTM. The comparison results indicate that the dropout and bidirectional LSTM layers do not affect the performance too much. However, the only exception is using dropout layer when predicting events in Syria. The possible reason is that the political situation is not stable and there are too much unexpected events. Therefore, dropout layer can improve the prediction performance by removing noises.[6]

Country	Model structure	NMAE of multiple attempts			
Australia	LSTM	0.11	0.12	0.11	0.12
	Bi-LSTM	0.12	0.13	0.11	0.12
	add dropout	0.12	0.13	0.13	0.13
Syria	LSTM	0.46	0.49	0.48	0.48
	add dropout	0.41	0.41	0.43	0.42

Train and predict using different structures (consider 30 day's history, m=10)

In addition, we tried to consider different length of history, and found that within a certain range, longer history can improve the prediction performance. Overfitting is the possible

reason why too long history leads to lower performance. In our model, we consider 30 day's history finally.

History length	20	30	40
NMAE	0.14	0.11	0.12

Results of considering different history length (use data of Australia, m=10)

Besides, for a certain model structure, we tested the influence of different hyper-parameters, including batch size, number of units in LSTM layer[2], epochs number, etc. We obtained the best result with batch size 32, unit number 20, "ReLu" activation function, and "Adam" optimizer. In general, 1500 ~ 2000 epochs are enough for training.

3.2 Hot event prediction

Currently, most text analysis focus on sentiment analysis, and classification (such as named entity recognition). Unlike these traditional text analysis, this task mainly deploys text analysis model to predict whether an event in a news will become a hot event by predicting the number of mentions of this news. Usually, if a news is hot, it might contain some special keywords and context, so text analysis can be used here to predict the popularity of events. [5]

3.2.1 LSTM model for text analysis

In this model as shown in the figure 4 below, each input is an embedded word from the preprocessed text. Embedding means replacing words with vectors and making similar words have close distance in the vector space, so that the vectors will have semantics meaning. The number of inputs is the length of the input text. The output is the predicted number of mentions.

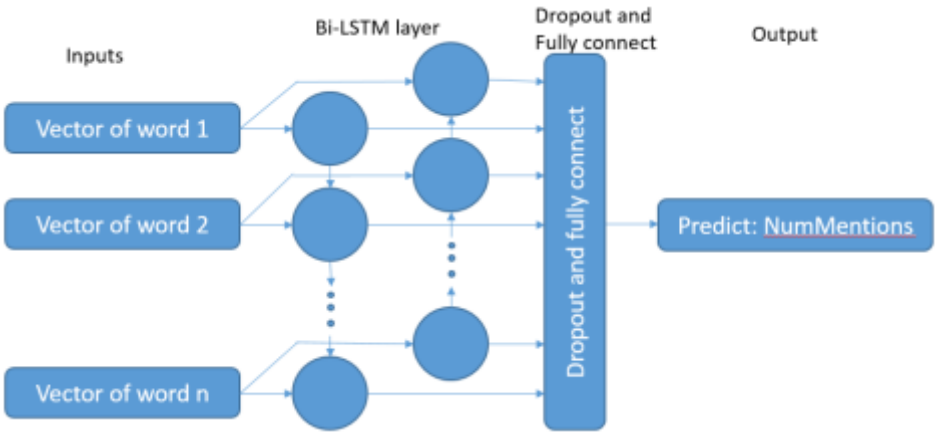

Figure 4 Text analysis model

In the hidden layer, we adopt bidirectional LSTM, which means adding one more LSTM layer, where neurons are connected in a reversed order. In the previous model used for time series, the ith neuron only receives data from the first to the (i-1)th neuron, which means only the previous data are considered. However, this is not enough for text analysis, because for a part in a text, both the context before and after this part could affect the analysis of this part. Using bidirectional LSTM can make the model be able to consider the full context. The Bidirectional method in Keras library is used to build this model.

3.2.2 Test results and analysis

[2] The first argument in Keras.layers.LSTM method. https://keras.io/layers/recurrent/#lstm

To evaluate our model, we use relative absolute error, which is |predict - truth| / truth, so that the scale of numbers can be ignored. Currently the lowest error is around 0.35. The main reason we think is that news is usually more redundant and noisy than some other commonly used sources in text analysis, such as movie reviews. [4]

Besides, we still obtained some useful conclusions about text analysis based on our experiment results under different conditions. The following results shows that removing some common words can improve the performance. These words nearly appear in all articles; hence, they cannot provide any help or reference for the analysis. However, removing too much will lead to lower performance. A possible reason is that some common words, such as huge, extreme, kill, and so on will significantly determine the popularity of a news. Finally, we remove the top 500 most common words.

How many common words to remove	don't remove	top 500	top 1k	top 3k
Error	0.41	0.35	0.38	0.82

The influence of removing common words

The following table shows that it is not necessary to consider all the words in the news, 500 words might be enough.

Cutting length	250	500	750	1000
Error	0.37	0.35	0.36	o.41

Results of different cutting length (top 500 common words are removed)

Experiments results also show that dropout layer can always improve the performance. It is probably because news articles are always noisy and contain some useless information such as backgrounds.

Besides, we also we tested the effects of different hyper-parameters in the model. We obtained our best result with batch size 100, unit number 30, dropout rate 0.5, "ReLu" activation function, and "SGD" optimizer. In general, 50-epoch training is enough, and more epochs can lead to severe overfitting.

3.3 State clustering

To find different states with the similar features, we utilized k-means clustering method by repeated calculating the distance between each state record (state record includes frequency of different event in 2013) and centroid. We tried to find the best performance of cluster with the lowest SSE in experiment by trying different number of centroids. Hence, the best clustering result in this project is 3 clusters in Australia, 8 clusters in China and 8 clusters in America.

4. Applications and Conclusions

In this project, we proposed two deep learning models for event occurrence and mentioned times prediction respectively. Firstly, a time-series prediction model was developed to forecast event occurrence in the different countries, which effectively predicted occurrence of multiple types of events simultaneously. Secondly, we adopted a text analysis model to predict the mentioned times of news related to events. Finally, the results of prediction are visualized through a map application and a web page, which are shown in the figures in appendix. In map application, users can easily select the date or event type they are interested

in the checkbox on the right panel to see the predicted trend of chosen events on the map. And in the web page, the news in last 3 days are ordered by the predicted popularity. Both applications, we think, are useful to be adopted in practice. This is because the clustered states are meaningful for government formulate policies in the same states group, while the news recommendation website can help to filter uninteresting new and thus save readers' time.

There are several directions for extension research. First, we would like to explore the relation of events pattern and other financial factors (e.g. GDP, average salary and inflation etc). Second, it would be an interesting future direction to analysis the inner association rules among various events in different countries. Third, due to the high noise of original text, it would be useful to create a background distribution to achieve rapid noise restraint in news texting data.

Through participating in the event prediction project, we had a strong point of the importance of group work, combination of practice and theory and practical implications of data mining. First, we have learnt the benefits of group working. Reasonable and scientific mission's allocation is the key to complete the whole project successfully when time and resources are limited. Then, by discussion and research, we had a better understanding on how to make theory learnt in textbook and practice coalescent. We also compared the advantages and disadvantages of different structure of prediction models during the process of implementing event prediction model. Finally, we get to know how to employ data mining skills to discover knowledge from dataset and provide valuable information for human beings.

Reference

[1] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," Neurocomputing, vol. 50, pp. 159-175, 2003.
[2] Karim, Fazle, et al. "LSTM Fully Convolutional Networks for Time Series Classification." *arXiv preprint arXiv:1709.05206* (2017).
[3] Leki, Ilona. "Twenty-five years of contrastive rhetoric: Text analysis and writing pedagogues." *Tesol Quarterly* 25.1 (1991): 123-143.
[4] Coulthard, Malcolm, ed. *Advances in written text analysis*. Routledge, 2002.
[5] Qi, Wei, et al. "Integrating visual, audio and text analysis for news video." *Image Processing, 2000. Proceedings. 2000 International Conference on*. Vol. 3. IEEE, 2000.
[6] Hamilton, James Douglas. *Time series analysis*. Vol. 2. Princeton: Princeton university press, 1994.

Appendix

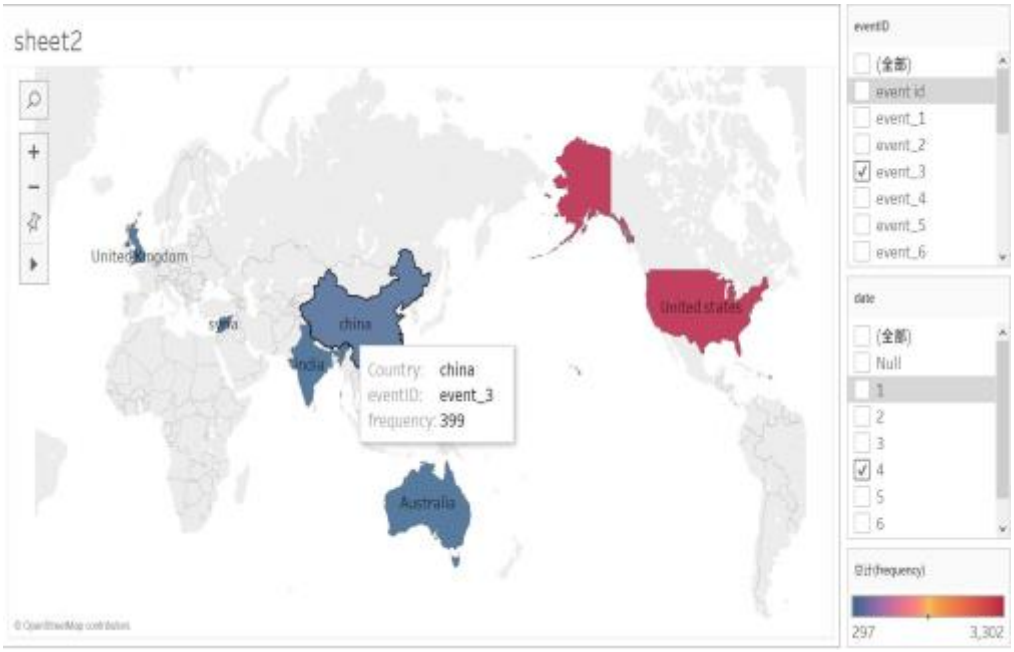

Figure 5 Map Application using Tableau

Figure 6 Three Clusters in Australia

Figure 7 Eight Cluster in China

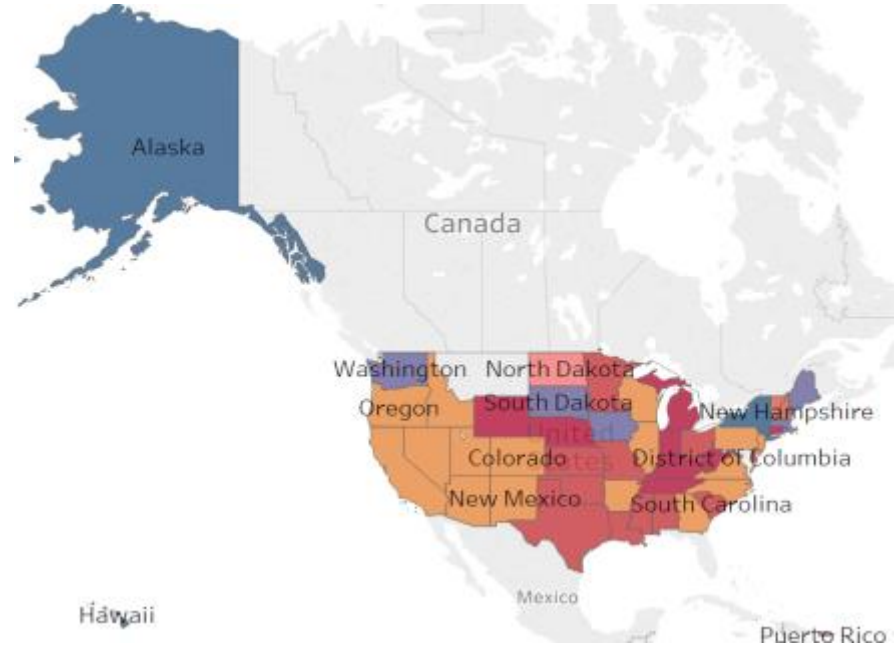

Figure 8 Eight Cluster in America

Figure 9 Web page for hot news recommendation