Journal of Sociology and Ethnology (2022)
Clausius Scientific Press, Canada

DOI: 10.23977/jsoce.2022.040703
ISSN 2616-2318 Vol. 4 Num. 7

# *Big Data-Based Analysis and Prediction of International Events*

**Bingnan Wang**

*College of National Security, People's Public Security University of China, Beijing 100038, China*

***Keywords:*** Big data, International event data, Deep learning, Encoder-decoder attention, News media tone

***Abstract*:** International event data can be seen as sensors that explain the interaction of countries. With the help of international event data, we can quantitatively analyze bilateral and even multilateral relations, which has very important reference value for stakeholders such as multinational companies and policy makers. This article explores the potential of the GDELT large database in the analysis and prediction of international event data. First, we introduced the GDELT database, data format and coding system of International Events Data Analysis. Second, we select all interaction events between China and other countries in the world from February 2020 to September 2021 to understand China's interaction with other countries and the trend of China's international evaluation through the analysis of conflict-cooperative events and media tone. Finally, we selected a more random "AvgTone" field for prediction, and proposed a prediction model based on the Encoder-Decoder Attention framework. After experiments, the model can still converge against data with a lot of noise, which proves the potential value of deep learning algorithms in international event data analysis.

## 1. Introduction

Predictive research is an important part of international relations research. Many international relations scholars agree that predictive research on international issues has the operability of social sciences. The core task of traditional forecasting research is to select appropriate conflict independent variables, through variable control experiments, and use statistical methods to discover the inevitable causal connection between independent variables and conflict outbreaks. Human society, as a complex system, often has many variables behind a simple social phenomenon[1]. The relationship between phenomena and variables is not a simple linear interaction, but a complex nonlinear activity, and there are also mutual influences between variables. It is very difficult to make a clear explanation of the complicated causality behind a certain social phenomenon.

The development of big data and data mining technology has raised the awareness of "relevance" in international problem forecasting research. Although big data technology has achieved good results in other fields, it has not yet received enough attention in the field of international relations. Therefore, it is necessary to explore the application of big data in the field of international relations and introduce new technical models to improve the accuracy of predictions.

## 2. Related Work

In the field of international relations, some scholars have used the data of GDELT and GTD to study the relationship between the content and frequency of information transmission and the outbreak of terrorist attacks. The content of information transmission is measured by the social sentiment value, that is, the Tone value of GDELT. The frequency of information transmission is calculated by the sum of the number of news reports and social posts related to the British government every day, and the signal content × signal frequency is used as a conflict vector to simulate real conflicts. Situation. Studies have shown that there is a correlation between the fluctuations of social emotions and the frequency of terrorist attacks. The lower the value of social emotions, the higher the probability of terrorist attacks. Some scholars use the event data in GDELT to propose a Multi-input LSTM-based international relations prediction model, which can be used to predict future conflicts and cooperation between countries[2].

In the field of conflict prediction, based on the event flow data of five countries in Southeast Asia, some scholars have used the Hidden Markov Model (HMM) to predict indicators related to national instability[3]. Some scholars use GDELT and machine learning models to study large-scale riots at the state and county levels in the United States, and to perceive and predict social unrest at the county level[4]. Some scholars have combined social media data, night light data and GDELT data volume to conduct research on the "Arab Spring"[5]. GDELT can quantify the development of conflict events in the "Arab Spring". Social media data and changes in night lights indicate the intensity of the conflict.

## 3. Data Analysis

### 3.1 GDELT Database

GDELT is a project initiated by Kalev Hannes Leetaru in 2012. It is a global, free and open rolling event database.With the support of Google Jigsaw, GDELT can detect portals, print media, and online media in more than 100 languages around the world in real time, and automatically extract important people, important organizations, geographic locations, news tone and other event elements in news media, and encode them Is the event data. The database uses the CAMEO coding system[6], updated once every 15 minutes. The event data provided includes both time and space dimensions.

GDELT stores data on Google's servers, and with the help of GoogleBigQuery, data can be extracted and preliminary exploratory analysis can be performed.With this service, users can use standard SQL to query and export data, and even perform analysis and modeling on the entire data set.

### 3.2 Data Analysis

Through the SQL query of GoogleBigQuery, we can get the event data we want arbitrarily. We screened all events with the Actor1CountryCode field of "CHN" from February 2020 to September 2021, a total of 295,996 pieces of data, representing all interaction events between China and other countries during this time period, and based on this analysis of China and Other interaction conditions.

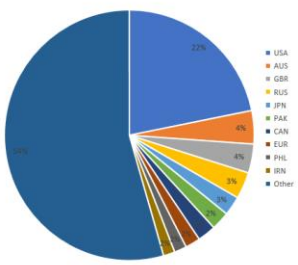


Fig.1 Proportion of Interactive Countries

The “QuadClass” field of GDELT divides events into four types based on conflict and cooperation: 1 verbal cooperation, 2 material cooperation, 3 verbal conflict, and 4 actual conflict. Through statistical analysis of the “QuadClass” field, we can roughly understand the conflict and cooperation between China and other countries.

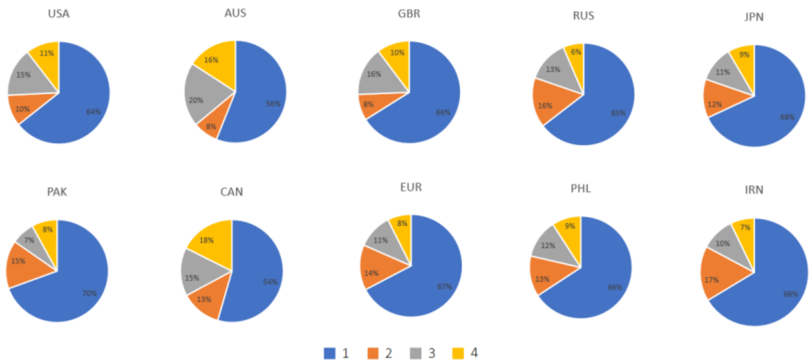


Fig.2 Proportion of Conflicts and Cooperation Incidents in Interactive Countries

By counting the “Actor2CountryCode” field of event participant 2, we can get the proportion of all countries that interacted with China from February 2020 to September 2021. Statistics show that the top ten countries with the most interactions with China are the United States, Australia, the United Kingdom, Russia, Japan, Pakistan, Canada, the European Union, Philippines and Iran, accounting for 54% of all countries. From this we can see that the countries that interact with China more often include several major developed economies, China’s strategic partners, and China’s neighboring countries. Therefore, with the aid of event data, we can see the status of a country in the international system structure and the country's diplomatic focus. By further analyzing the proportion of conflicts and cooperation events in each country, we can also explore the specific interactions between China and each country.

## 4. Model

### 4.1 System Overview

In the past, GDELT-based forecasting studies were mostly about the number of events of a certain type. Combining with the powerful fitting ability of neural network, in this article we try to predict a more random field “AvgTone”. Based on the previous exploratory analysis of GDELT data, we propose a media tone prediction model based on the Encoder-Decoder Attention framework. The sequence of events in the previous month is input to the model to predict the tone of the media in the next week. In our proposed news media tone prediction model, it mainly includes two parts: (1) event encoder, used to extract the features of the event sequence of the previous month, (2) attention layer, the influence on the event sequence of the previous month Calculate the degree to get the weight score (2) Event decoder, according to the event features extracted by the encoder and the context vector generated by the attention layer to generate the media tone of the next week.

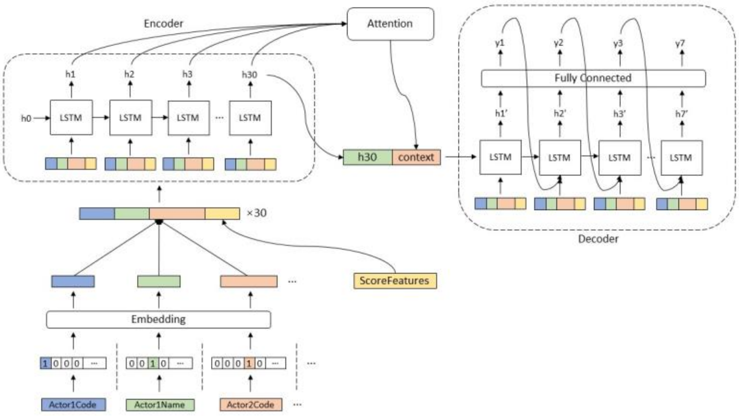


Fig.3 Model Overview

## 4.2 Model Architecture

### 4.2.1 Embedding Layer

In machine learning tasks, one-hot encoding is usually done for discrete features that have no meaning in size. The one-hot encoding method is to use N-bit status registers to encode N states, and each state has an independent register bit. For a feature, if it has n possible values, it becomes n binary features after one-hot encoding.

Although one-hot encoding solves the problem of processing category features, it also has a series of shortcomings. Since only one bit in each status register is active, it will cause the one-hot vector to be too sparse and excessively occupy computing resources. Therefore, we need to perform the calculation after embedding each one-hot vector. The embedding layer in the model can be understood as mapping data from a high-dimensional space to a low-dimensional space. In each embedding layer, matrix multiplication is performed on the one-hot vector. The calculations performed in the embedding layer are as follows:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix} \tag{1}$$

The specific process of embedding is as follows: First, a vocabulary is constructed for each category feature, which contains all the value types in the category feature, and the vocabulary will assign an index value to each value. For example, we have a total of 295,996 data samples, and the type of value contained in the Actor1Code field is 79. Therefore, the vocabulary length of Actor1Code is 79, and the index value is [1,2,3, ..., 79]. Therefore, the 295,996 values in the Actor1Code field can all be represented by integers from 1 to 79. According to the index value of Actor1Code, the deep learning framework can automatically create a one-hot matrix of 295996×79. This one-hot matrix is subjected to a matrix multiplication operation with an embedded query matrix with a dimension of 79×M, and a matrix with a dimension of 295996×M is obtained. After the above calculation, the original 79-dimensional feature vector is reduced to an M-dimensional vector (M<79).

### 4.2.2 Lstm Encoder

Recurrent Neural Networks(RNNs) are proposed in 1980s and in principle can creat and process memories of arbitary sequence of input patterns. RNNs are kinds of neural networks that are specially used for processing sequential data like x^((1)),...,x^((t))just as Convolution Neural Networks(CNNs) for grid data in special. Due to the inevitable defects of RNNs, some improved versions of RNNs are proposed. The Long Short-Term Memory Network(LSTM)was proposed to solve the defect of long-term dependency of RNNs[7]. Long-term dependencies refer to the fact that the state of the current system may be affected by systems that existed a long time ago, a problem that RNNs cannot solve. In general, LSTM performed better in memory of long sequences than the normal RNN. Considering that there is a certain autocorrelation in that tone of news media for a period of time, We choose LSTM as the feature extractor of the event sequence in the part of model encoder, which is used for extracting the time-dependent features in the event sequence of the previous month. All RNNs contain chain repeat units of neural network. Unlike RNN, there are five components in an LSTM unit: an input gate i_t , a forget gate f_t , an output gate o_t, a memory cell c_t, and a hidden state h_t. They are all vectors in R^k, where k is the dimension of hidden state. Formally, given a time series {x_1 , x_2, ...,x_T} with x_t ∈ R^m, each step in LSTM's recursive process can be defined as a collection of functions as follows:

$$\begin{aligned}
i_t &= \sigma(W_i[h_{t-1}; x_t] + b_i) \\
f_t &= \sigma(W_f[h_{t-1}; x_t] + b_f) \\
o_t &= \sigma(W_o[h_{t-1}; x_t] + b_o) \\
g_t &= \tanh(W_g[h_{t-1}; x_t] + b_g) \\
c_t &= i_t \odot g_t + f_t \odot c_{t-1} \\
h_t &= o_t \tanh(c_t)
\end{aligned} \tag{2}$$

Where x_t is the input of current time step t,h_(t-1) is the hidden state of last time step t-1, W_i,W_f,W_o,W_u∈R^(k×(k+m)) are the weight matrices , b_i,b_f,b_o,b_g∈R^k are the bias vectors. ⊙ represents element-wise mutiplication, σ and tanh denotes sigmoid function and hyperbolic tangent function. The functions are as follows:

$$\begin{aligned}
S(x) &= \frac{1}{1+e^{-x}} \\
\tanh x &= \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}
\end{aligned} \tag{3}$$

It can be seen from the mathematical formulas of the LSTM unit that when the input x_t of t moment enters the LSTM unit, the input gate i_t controls the input value that needs to be updated in the memory cell c_t of the current moment, the forgetting gate f_t controls the update of the median value in the memory cell c_(t-1) of the previous moment, and the output gate is responsible for o_t filtering the stored value in the memory cell c_t of the current moment and generating the hidden state h_t, which enters the update of the next cycle together with x_(t+1) of the current moment.

Therefore, given an input of a sequenc data {x_1 , x_2, ...,x_T}, x_t∈R^m , the output of LSTM network can be seen as {h_1 , h_2, ...,h_T}, h_t∈R^k.

### 4.2.3 Attention Layer

The Attention mechanism has been widely used in various deep learning tasks and has achieved very good results. The intuition of the Attention mechanism is that the importance of different parts of the input sequence is different, and the gating unit of LSTM can also be regarded as a saliency-based Attention. At present, most of the Attention mechanisms are applied in the Encoder-Decoder framework. In RNN, the application of Attention is to multiply the calculated attention weight with the hidden state generated by the RNN. According to existing research, many Attention mechanisms have been produced. Minh-Thang Luong studied the application of Attention in machine translation tasks, and proposed Global Attention and Local Attention[8]. Local Attention only applies Attention to fixed-size windows each time, so it can save computing resources compared to Global Attention, but the selection of the time window size will affect the calculation results. In order to maximize the effect of the model, we select Global Attention as the Attention layer of our model.

The Global Attention mechanism will consider all the hidden states of the Encoder when generating the context vector c_t. h_t represent the current hidden state while h˜_s denotes the hidden state of all time steps of the input sequence.

After calculating the similarity between h_t and h˜_s, a variable-length alignment vector is generated, and its dimension is equal to the number of time steps belonging to the sequence:

$$\begin{aligned} a_t(s) &= align(h_t, \tilde{h}_s) \\ &= \frac{exp(attn(h_t, \tilde{h}_s))}{\sum_{s'} exp(attn(h_t, \tilde{h}_s))} \end{aligned} \tag{4}$$

There are three main methods for calculating attn(x):

$$attn(x) = \begin{cases} x &= h_t^{\top} \tilde{h}_s & dot \\ y &= h^{\top} W_a \tilde{h}_s & general \\ z &= v_a^{\top} tanh(W_a[h_t; \tilde{h}_s]) & concat \end{cases} \tag{5}$$

In the results of Minh-Thang Luong's research, dot is better for global, so we choose dot calculation method in our model. The specific method is to calculate the similarity between the hidden state on the 30th day (32-dimension) and the hidden state of the first 30 time steps as a dot-product, and then calculate the attention weight through softmax, and hide the weight with each time step of the RNN Multiply the states to get the final context vector (32-dimension). Finally, after splicing with the context, it is input into the decoder as the initial hidden state (64-dimension) of the decoder.

### 4.2.4 LSTM Decoder

The structure of the Decoder part of the model is similar to that of the Encoder part. We also select LSTM to decode the features extracted by the Encoder part. The difference is that in the Encoder part, we initialize h_0 at the 0th time step of the LSTM to an all-zero vector. In the Decode part, the initial hidden state of LSTM is the splicing of h_30 and context vector. After the event occurs, the encoding of the event feature has been determined, and the tone of the media report is calculated after the media report. Therefore, we use the event feature as the input of the Decoder, which is the same as the Encoder part, and these attributes also need to be embedded. For the hidden state generated by LSTM, we add a fully connected layer to process it, and map the hidden state learned by LSTM to the sample label space corresponding to the label.

The input of the Decoder part is the event feature after embedding. Since “AvgTone” is the part we want to predict, we initialize the first input “AvgTone” field to zero, stitch it with other embedded features, and input it into the decoder. After the first time step’s hidden state h_1 of LSTM is processed by the fully connected layer, the predicted value y_1 of the first day is obtained, and then we splice y_1 with the event attributes of the second day as the input of the second time step of LSTM , And so on to get the predicted values of “AvgTone” in the next 7 days.

## 5. Experiment

First of all, we first perform one-hot encoding on the category features of the event. The specific operation is: use a counter to count the number of occurrences of each category feature value. This step can calculate how many types are in each category feature Feature value; use the natural language processing module “torchtext” in Pytorch to construct a vocabulary for each category feature, and automatically create an integer index for each value, the index indicates the location of the one-hot status register activation; finally, each category feature Integer index for splicing, python can automatically create a one-hot matrix of the corresponding dimension according to the index.

Second, the dimensions of the data must be processed to meet the requirements of the model. Because we want to use the events of the previous 30 days to predict the tone of the media in the next 7 days, we divide the 295,996 pieces of data into 9866 sequences of length 30, discard the last 16 pieces of data, and use them as the input part of the model with a dimension of $9866 \times 30 \times 7$. Then we sample the 7 pieces of data after 30 pieces every 30 days as the input and output of the Decoder part of the model, with a dimension of $9866 \times 7 \times 7$.

Then, embedding features of each event category. We have selected 7 category features, there are a total of 7 embedding layers at the input. Suppose the dimension of the input one-hot matrix is $295980 \times N$, and the dimension of the query matrix is $N \times M$. For the calculation method of M, we set it to N^(1/4), and then round down N^(1/4) to get the final value of M. After we embed each category feature, we splice each category feature and the numerical feature of the event to obtain the event sequence representation on the Encoder side.

In the Encoder part of the model, the dimension of the output vector $h_{30}$ is 32, and the context dimension calculated by Attention is 32. The two parts are spliced into a vector with a dimension of 64 and input into the Decoder part of the model. We randomly sample 70% of all data as the training set and 30% as the test set. We select the Adam algorithm as the model optimizer, the learning rate is set to 0.0001, the error calculation method is selected as MSELoss, the model is trained for 200 epochs, and the MSELoss of the training process The decline is as follows:

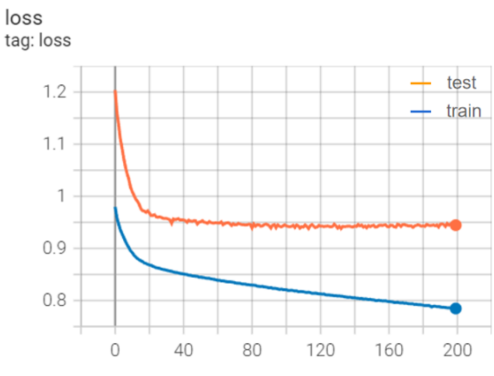


Fig.4 Mseloss

From the figure, we can see that when the model is trained to the 20th epoch, the MSELoss of the test set drops to the lowest point of about 1.0, after which there is basically no change, and the MSELoss of the test set can continue to converge. The reason is that after the 20th epoch, the model has been overfitted on the training set, so the MSELoss on the test set no longer drops.

## 6. Conclusion

This research focuses on the application value and potential of the GDELT large database in international event data analysis, and sorts out the data types, historical development and application of international event data analysis. The GDELT large database selects China from February 2020 to 2021. All interaction events with other countries in September, analyzing the interaction events between China and other countries in the context of the epidemic and China's international media evaluation. Select the top ten countries with the highest interaction frequency with China, analyze the conflicts and cooperation between China and these countries during this period through analysis of conflict-cooperation events.Through real-time monitoring of a country's event flow and analysis of whether there are abnormal fluctuations, the occurrence of major events within the country can be monitored in real time.

Based on the above data exploration and analysis, we also proposed a media tone prediction model based on Encoder-Decoder Attention. The model encoder extracts the features of the event sequence of the previous month, and then calculates the attention weight of the hidden state of the encoder at each time step through the Attention mechanism, and splices the generated context vector with the output of the encoder at the last time step. , As the initial hidden state of the decoder to generate the media tone of the next week. Experiments show that the MSELoss could reach 0.95.

## Acknowledgement

The present study was endorsed by the National Social Science Foundation of China (Grant No. 21BXW066), Fundamental Research Funds for the People's Public Security University of China(Grant No.2021JKF207).

## References

*[1] Qingling Dong, "Big Data Security Situation Awareness and Conflict Prediction". Chinese Social Sciences, vol.06, no.2, pp.172-182, 2016.*

*[2] Peng Chen, Adam Jatowt, and Masatoshi Yoshikawa, "Conflict or Cooperation? Predicting Future Tendency of International Relations". in Proceedings of the 35th Annual Acm Symposium on Applied Computing, pp923–30, 2020.*

*[3] Menglan Ma et al., "Does Ideology Affect the Tone of International News Coverage?". International Conference on Behavioral, Economic, Socio-cultural Computing (BESC), pp1-5, 2017.*

*[4] Divyanshi Galla and James Burke, "Predicting Social Unrest Using GDELT". International Conference on Machine Learning and Data Mining in Pattern Recognition, pp103-16, Springer 2018.*

*[5] Noam Levin, Saleem Ali, and David Crandall, "Utilizing Remote Sensing and Big Data to Quantify Conflict Intensity: The Arab Spring as a Case Study". Applied Geography, vol94, no.3, pp1-17, 2018.*

*[6] Yilmaz, Philip Schrodt, and Deborah Gerner, "Conflict and Mediation Event Observations (CAMEO): An Event Data Framework for a Post-Cold War World". Security and Conflict Management, vol. 4, pp287-304, 2008.*

*[7] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory," Neural Computation 9, vol.12, no. 8, pp1735–80, 1997.*

*[8] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective Approaches to Attention-Based Neural Machine Translation". ArXiv: 1508. 04025 [Cs], 2015.*