CS6604 Project Report

Project Name:
IDEAL Computational Linguistics Prototype
-Unsupervised Event Extraction from News and Twitter

Instructor: Edward A. Fox
Client: Mohamed Magdy

Group members:
Tianyu Geng (Volunteer)
Ji Wang (Auditing)
Wei Huang
Xuan Zhang

News event extraction: by Xuan and Wei
Twitter event extraction: by Ji and Tianyu

Date: May, 8th, 2014
Virginia Tech, Blacksburg, VA

Table of Contents

List of Figures

List of Tables

0. Abstract

Living in the age of big data, we are facing massive information every day, especially that from the mainstream news and the social network. Due to its gigantic volume, one may get frustrated when trying to identify the key information which really matters. Thus, how to summarize the key information from the enormous news and tweets becomes essential. Addressing this problem, this project explores the approaches to extract key events from newswires and Twitter data in an unsupervised manner, where the Topic Modeling and the Named Entity Recognition have been applied. Various methods have been tried regarding the different traits of news and tweets. The relevance between the news events and the corresponding Twitter events is studied as well. Tools have been developed to implement and evaluate these methods. Our experiments show that these tools can effectively extract key events from the news and tweets data sets. The tools, documents and data sets can be used for educational purpose and as a part of the IDEAL project of Virginia Tech.

1. Introduction

Nowadays, the explosion in the volume of digitized textual content online has threatened to overwhelm human attention. Every day, hundreds of Megabytes of news stories are being dumped into the news archives of the major news agencies, containing many uninteresting or trivial news. Clearly, it is almost impossible for people to absorb all pertinent information from such vast amounts of news articles in a timely manner. Thereby, event extraction, which attempts to identify the key "events" by exploring and analyzing the content of textual materials, has emerged as a promising research area to alleviate the information overload problem.

As part of the Integrated Digital Event Archive and Library (IDEAL) Project, our work implemented a program to automatically extract and organize hot events from a given set of text-based news webpages published during a given time period. We segment the input text into textual units (e.g., title and paragraph), parse each relevant textual unit, figure out what topics it covers, and choose a set of keywords (who, what, when, where) to summarize this event. We also developed a tool to extract text content from archived news web pages, which can facilitate online news collecting.

In another aspect, social networks, such as Twitter, have become one of the most important platforms to publish information and share thoughts during many events. Therefore, the events reported by tweets are being studied by researchers, such as [1] [2]. We get involved in this research by collecting 130,000 tweets regarding the "Ukraine Crisis" story, and exploring the method to extract events from them.

Compared to the existing research [3] [4], we are trying to design a new method to study the relationship between the Twitter events and news events. The news and tweets related with the same story, "Ukraine Crisis", were collected and employed in the event extraction study.

Our project has focused on the *unsupervised event extraction*: we explored various methods regarding the topic modeling and named entity extraction of the event extraction, without the dependence on the annotated data. This project produced comprehensive and valuable resources, including documents, code, datasets, etc., to the students in the new course "Computational Linguistics". We also provide the prototype for extract summative information from specified collection, which can serve as pipelines of the IDEAL project.

The user manual of the tools developed, the developer manual for extension purpose, the lessons learned from this project, the experiments and assessments are included in this report.

2. User’s Manual

2.1 News Event Extractor

The latest release has been tested for compatibility with Microsoft Windows 7 and CentOS 6.4.

2.1.1 Using News_Event_Extractor

- Download archive file *NewsEventExtractor.zip*, and expand it into some directory;
- Make sure Java VM (V1.7.0+) installed, and the “java” command is in the “PATH” environment variable
- Modify the configuration file “*config.properties”* according to the actual environment;
- Run the batch files "*start.bat*" or “start.sh”, to extract events set from specified documents;

2.1.2 Configuring News_Event_Extractor

The configuration file “*config.properties”* contains three variables: NEWS_DIR, TOPIC_NUMBER, and WORD_PER_TOPIC.

```
NEWS_DIR : D:\Test\UkraineCrisis
```

NEWS_DIR should be pointed to the directory where news text files are saved.

```
TOPIC_NUMBER : 10
```

TOPIC_NUMBER allows to specify the number of topics (events) extracted from news.

```
WORD_PER_TOPIC  : 7
```

WORD_PER_TOPIC allows to specify the number of word in each topic.

Then start the program by running the batch files

- start.bat (Under Windows)
- start.sh (Under Linux, Unix, and Mac OS)

Attention: News with timestamp is needed. Every piece of news should be a text file which is placed under a disk path ending in the “YYYY/MM/DD/newstitle.txt” pattern.

2.2 Twitter Event Extractor

1) LDA topic modeling:

python trainLDA.py <tweets_source_file>
python trainLDA.py <tweets_source_file>

2) NER process:
python assignNER.py <tweets_source_file>

3) Event Creator:
without FP Growth: *python fillEvent.py*
with FP Growth: *python fillEvent_FP_Growth.py*

3. Developer's Manual

3.1 Concept Map

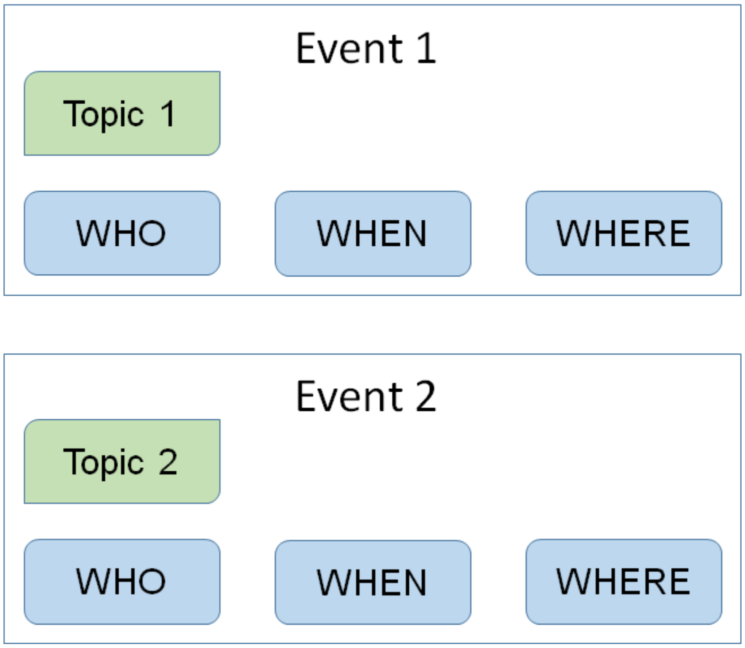

Figure 1. Definition of Event

In this project, the event is defined as a tuple of {Topic, Named Entities} for both news and tweets. Every event has a topic, which is a word set describing the basic story of this event. Furthermore, some named entities are also involved, to tell the people, the location, and the time of this event.

3.1.1 Architecture of News Event Extractor

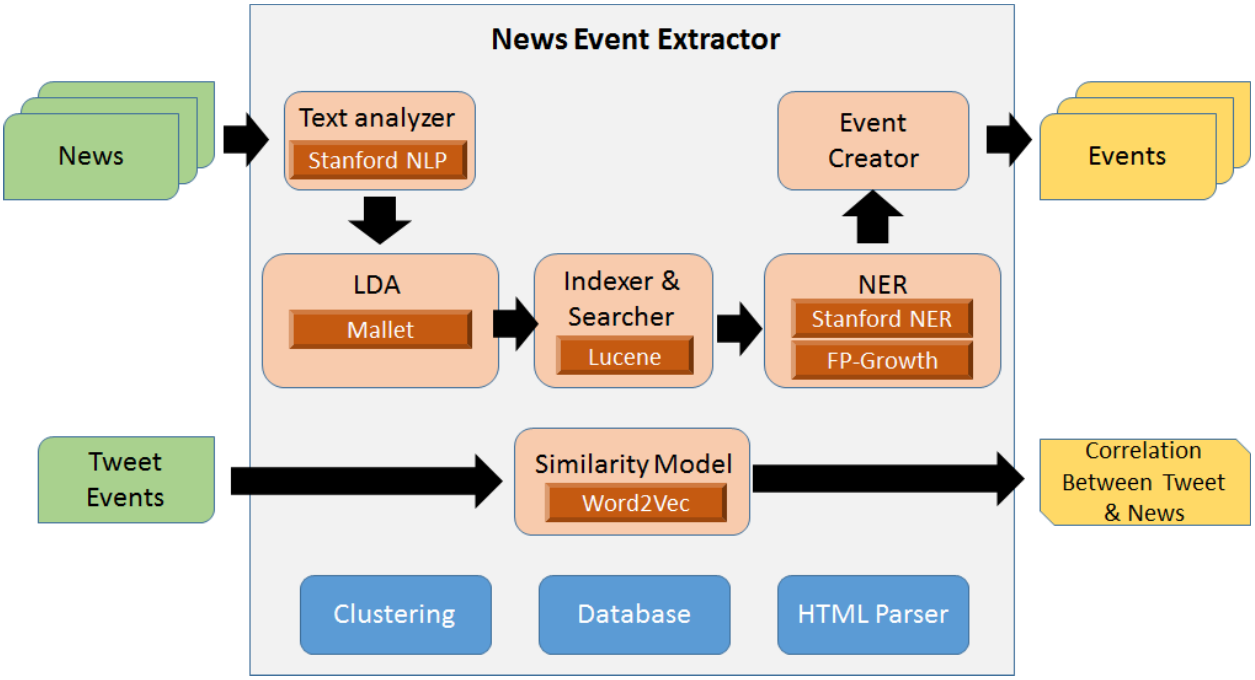

Figure 2. Architecture of the News Event Extractor

The architecture of the News Event Extractor is demonstrated by the figure above. Two important functions have been implemented, the event extraction, and the event similarity calculation.

To extract events from news articles, first, the text analysis (tokenization, stemming, stop-word removing, etc) is performed to the imported news text by the Text Analyzer, which is implemented based on the Stanford NLP [5]. After that, topics are extracted by the LDA module, which relies on the Mallet tool [7]. Since a topic, a word array with usually 7~10 words, is too short to extract the named entities, we need to find the named entities (people, location, and time) which are related with the event from the news paragraphs which are closely relevant with the topic. Thus, the Apache Lucene [8] is used to index all the news paragraphs, and to search the paragraphs which are relevant with a particular topic. The named entities (who, where, and when) are extracted from these paragraphs by the NER module, which has applied the Stanford NER tool [6]. These entities are categorized into 3 types and ranked by term frequency, in order to find key entities regarding an event. In addition, the important named entity combinations are also identified to help users understand the event. The FP-Growth algorithm [14] [11] is utilized to identify these combinations. Finally, an “Event Creator” component produces events based on the extracted topics and corresponding named entities.

In term of the event similarity model, the Google Word2Vec [15] [9] tool has been used to produce a vector set for every event. The distance between two events is measured by a similarity model, which calculates the distance between the centroids of two spheres formed by the two vector sets, in a hyper dimension space.

Some components like the database, weka clustering [10], and HTML parser offer fundamental functions to the system.

3.1.2 Architecture of Twitter Event Extractor

Same as the pipeline of the News Event Extractor, the Twitter Event Extractor can also divided into four steps: Text Analyzer, LDA, NER and Event Creator.

- Text Analyzer: the text analysis (tokenization, stemming, stop-word removing, etc) is performed to the tweets' text by the Text Analyzer, which is implemented based on the Stanford NLP.
- LDA: the LDA component is to perform topic modeling process for tweets content. The Python library we used is Gensim LDA component [12]. Firstly, we use LDA component to create the LDA model by the whole tweets content. Then, for each tweet, we use the model we built to label topic to each tweet.
- NER: The NER component is for the Name Entities Recognition (NER). In this part, we used PyNER python library [13] as the interface of Stanford NER service. Our NER parsing process used 7 classes name entities. They include Time, Location, Organization, Person, Money, Percent, and Date.
- Event Creator: after we got the NER results and topic labels of each tweet, we can generate the event. For each event, it contains 7 keywords from LDA topics, "When" component and "Where" component. We get "When" component from "Person" and "Organization" labels of NER result and "Where" component from "Location" labels of NER result.

3.2 Data Set

For the news data, we gathered *4084* news regarding the "Ukraine Crisis" story from the Reuters website. The time period of these news is 2 months (Feb, 18th 2014 to Apr, 18th 2014).

For twitter data set, the format is same as that in IDEAL tweets collection http://spare05.dlib.vt.edu/. Same as news article dataset, the time period of these tweets is also 2 months (Feb, 18th 2014 to Apr, 18th 2014).

3.3 Inventory of Files

3.3.1 Inventory of News Event Extractor

Table 1. Source Files of News Event Extractor

Package	File	Description
clustering	EMCluster.java	Clustering specified documents with EM algorithm

	KMeansCluster.java	Clustering specified documents with K-Means algorithm
console	EventMain.java	The main program that takes massive news texts as input, and outputs the events (coherent key words set)
	FileFilter.java	The main program that filter out irrelevant files, based on the specified key words set
	ImportNewsToDB.java	The main program to import news files into SQL database
	LDAConsole.java	The class to test the JGibbLDA
	LDAonNewsBodyClusters.java	The main program to extract topics from the news clusters which are clustered by the news body
	LDAonNewsTitleClusters.java	The main program to extract topics from the news clusters which are clustered by the news title
	testNamedEntities.java	Main function to call NERUtil.extractNamedEntities
	TFIDF.java	Function to calculate TF-IDF for specified documents
constants	ClusterConstants.java	Define constant *CLUSTER_NUM*
	FeatureConstants.java	Define constant *TOP_N_FEATURES*
	GeneralConstants.java	Define constants *WORKING_DIRECTORY* and *WORD2VEC_MODEL_PATH*
database	MysqlConnector.java	Function to get connection from MySQL
	SQLCMD.java	A SQL commander which supports "update" and "query"
encoding	EncodingUtil.java	Convert the specified file from one encoding format to the other
	UnicodeUtil.java	Convert plain text to unicode, or convert unicode to oriental language (e.g.Chinese)
event	EventExtractor.java	Create an event based on the specified topic and related documents, from which named entities can be extracted
fpmin	FPComparator.java	Comparator function for FrequentPattern
	FpGrowth.java	Implement FP-Growth Algorithm
	FpNode.java	Implement data type *FpNode*
	FrequentPattern.java	Implement data type *FrequentPattern* type
	Reader.java	Read the file, then outputs the data as character string

		or matrix
htmlparser	TextExtract.java	Extract the main content of given html file
	UseDemo.java	Main function to test *TextExtract* class
LDA	Filter.java	Some comparator functions used in *LDAUtility*
	LDAUtility.java	Some common functions related to LDA
	TopicModel.java	Function to extract topics set from specified documents based on the Mallet tool
	Word.java	Implement data type *Word*
model	Distance.java	Implement data type *Distance*
	Document.java	Implement data type *Document*
	Event.java	Implement data type *Event*
	NamedEntity.java	Implement data type *NamedEntity*
	TermMetric.java	Implement data type *TermMetric*
	Topic.java	Implement data type *Topic*
	TopicSet.java	Implement data type *TopicSet*
ner	NERConstants.java	Define constants used in package “ner”
	NERUtil.java	Functions to get the frequent named entity combinations from the specified file
preprocess	DocumentProcess.java	Some common functions related with the processing of Document objects
	StanfordLemmatizer.java	Some common functions related to lemmatization
search	LuceneSearcher.java	Extend Lucene to index all the news paragraphs, and to search the paragraphs which are relevant with a particular topic
	SearchConstants.java	Define constants used in package “search”
	Searcher.java	Define interface *Searcher*
utility	Ansi2Utf8.java	Utility class: Convert text file from ANSI to UTF-8
	DateUtil.java	Some common functions related to Date
	FileUtil.java	Some common functions related to File

	MathUtil.java	Some math-related functions
	RegularExpression.java	Some common functions related to regular expression
	TermMetricComparator.java	Comparator function for term weight
	TermMetricUtil.java	Some common functions related to TermMetric
	TextUtil.java	Some text-related functions
	TopicComparator.java	Comparator function for topic probability
word2vec	Word2VEC.java	Extend Google Word2Vec tool to produce a vector set for every event
	Word2VecUtil.java	Utility class: For every item in the list1, get the most similar item from list2, and return them in a list

3.3.2 Inventory of Twitter Event Extractor

Table 2. Source Files of Twitter Event Extractor

Package	File	Description
idealtwitter	assignNER.py	extract name entities from tweets
	assignTopics.py	assign LDA topic modeling label to each tweet
	fillEvent.py	fill event by LDA and NER results
	fillEvent_FP_Growth.py	fill event by LDA and NER results. then use FP Growth to pick the most frequency combination of event.
	trainLDA.py	use LDA to get topic modeling.
GoogleAlertRSSFilter	rssFilter.py	extract urls from google news alert.

4. Lessons Learned

4.1 Timeline/schedule

- Feb. 21 - Literature review
- Mar. 4 - Text preprocessing and LDA
- Mar. 6 - Midterm presentation
- Apr. 7 - LDA improvement
- Apr. 13 - Named entities extraction
- Apr. 27 - Event composition
- May. 1 - Final presentation
- May. 8 - Final report

4.2 Problems

4.2.1 News topic modeling

The *overlap* and the *noise* are the biggest problems for topic modeling. Overlap emerged in more than 60% of extracted topics, when we were applying LDA to the entire news articles of a small data set (around 200 news). Much noise appeared in the topics as well.

One of the important reasons is, there's too much noise within the body of the news article. Within a news article, some paragraphs may not be quite relevant with the theme of that article. For example, in a news article which reported the referendum of Crimea, most of the paragraphs were introducing the location and history of Crimea, which is noise in term of the main theme. Another reason is the size of the data set: it's hard to obtain reasonable topics by applying LDA to a small data set.

4.2.2 Named entity extraction

1) Token recognition

The outputs provided by Stanford NLP (CRFClassifier class) simply chop up words just by detecting white spaces, which will annotate “April”, “1”, “ , ” and “2014” as four different DATE-related tokens, or divide “White House” into two different ORGANIZATION-related tokens. However, what we expected is some more meaningful results, for example, "San Francisco" should have been a single token.

2) Pattern combination

In order to find out the relationship between named entities, we used FP-Growth algorithm to get all different possible permutations of the extracted named entities. As might be expected, shorter combinations usually have higher frequency, but contain less information. For example, “Obama; U.S.” may appear much more times than “March 5, 2014; Obama; Boston”, but obviously the latter one can play more important role in the event summary. Another problem is about the redundant patterns. For example, “March 5, 2014; Obama”, “March 5, 2014; U.S.”, “March 5, 2014; Obama; Boston”, and “March 5, 2014; Boston; Obama” should be merged into a single record like “March 5, 2014; Obama; Boston; U.S.”.

4.2.3 Tweet event extraction

Both LDA and NER have the challenge in tweets event extraction. The major reason is the length of tweet. As we known, each tweet only has 140 characters, including text, special symbol, and URL link(s). It means that there are few words in tweets than that in news article’s paragraph. The length of text will reduce the accuracy of topic modeling and name entities recognition.

For LDA topic modeling, assigning a topic to a message which has less than 140 characters is a not easy task. At the same time, for NER extraction process, each tweet only have 140 characters. Based on our experiment, it is very hard to extract enough “Where” and “Who” components from one tweet.

4.2.4 Link tweet events to news events

The relationship between the events emerging in mainstream news and those in Twitter remains as a research question explored by few. The consistency, similarity, and difference between the two types of media are worthy studying. In this project, we focus on the similarity between the news events and Twitter events.

4.3 Solutions

4.3.1 News topic modeling

Addressing the overlap and the noise problems, we changed the objects of the topic modeling. Instead of the entire news articles, we took the news titles as the target of LDA. That is because the news titles are very good summary of the news, with little noise. In addition, we substituted the small data set (*200* news) with a much larger one (over *4,000* news of the Ukraine Crisis).

Much better topics were extracted by this means. Among the extracted topics, the number of duplicate topics is no more than *20%*. Although there was still duplicated words among various topics, most of them were named entities. This is reasonable because there were no big change with the people and the location of among different events. For example, many events in the

“Ukraine Crisis” story were relevant with Vladimir Putin. Thus, Putin appeared in multiple topics. However, we can still distinguish these topics when we try to connect all the words inside them, especially from some action words. Furthermore, the noise inside each topic was significantly reduced as well.

Some other approaches have also been tried to improve the accuracy of topic modeling.

- Cluster the news by their titles before LDA. No good result. News titles are short texts, thus the title vectors are too sparse to be clustered accurately.
- Cluster the news by entire article before LDA. No good result. There is too much noise in the news body, which deteriorates the clustering result.
- Split the entire dataset into datasets with shorter periods. Topics with finer-granularity obtained.However, more noise emerged compared to topics from the entire data set.

4.3.2 Named entity extraction

1) Token recognition

The output of CRFClassifier is a nested list *List<List<CoreLabel>>*, where each CoreLabel represents a single word with ancillary information such as category. Based on this raw result, we scan through each CoreLable node, and compare the category of current node to the previous one. If two adjacent nodes have same category, we replace the *Word* field in former one with the string combination, remove the latter, and update the linked list.

2) Pattern combination

First, we decide to use “When” as a must-have join point to merge interlinked patterns, because the news is unique to timeliness and we believe timeline will be the most suitable way to visualize our event summary.

Then we set up a rule filter to make sure that only the patterns which contain “When” and at least one “Where” or “Who” can served as valid input for next step. Obviously, patterns like “March 5, 2014” or “Wednesday” are useless, since they cannot contribute extra information to enrich the keywords combination. In contrast, the patterns such as “March 5, 2014; Boston” or “March 5, 2014; Obama; U.S.” can provide some more details regarding this event.

After that, we create HashSet for each time point word, and collect all other high-frequency words which generally found together with this date. We insert LOCATION right after the primary key (DATE) and append PEOPLE or ORGANIZATION to the end of the set. In this way, we can obtain a keywords set for each time point of specific event, where the words are arranged by “When, Where, Who” without repetition.

Finally, we pick out the key dates whose occurrence frequency are larger than the threshold. The threshold varies from event to event, and is dynamically adjusted based on the maximum number of occurrences of combinations under this event.

In essence, with the above steps we can significantly reduce the number of combinations so that extract the most important parts from the input texts. The compression ratio is usually about two orders of magnitude. In some cases, we even have observed that over 6000 raw combinations were condensed into a single keywords set which makes sense for the corresponding topic.

4.3.3 Tweet event extraction

In order to increase the accuracy of LDA topic modeling results, we align tweets in every 1 min and got the enough text for LDA training and inference. For the NER part, we firstly extract name entities from each tweets. And then, we align all the tweets under same topic, get the larger name entities pool. At last, we use FP Growth to pick the most high frequency combination of “Where” and “Who” from the name entities pool of same topic. For the extraction process, we back race all the event to the raw tweets.

4.3.4 Link tweet events to news events

Given the events extracted from news and tweets, we want to identify the similarity between them. In particular, the distance between a news event and a Twitter event.

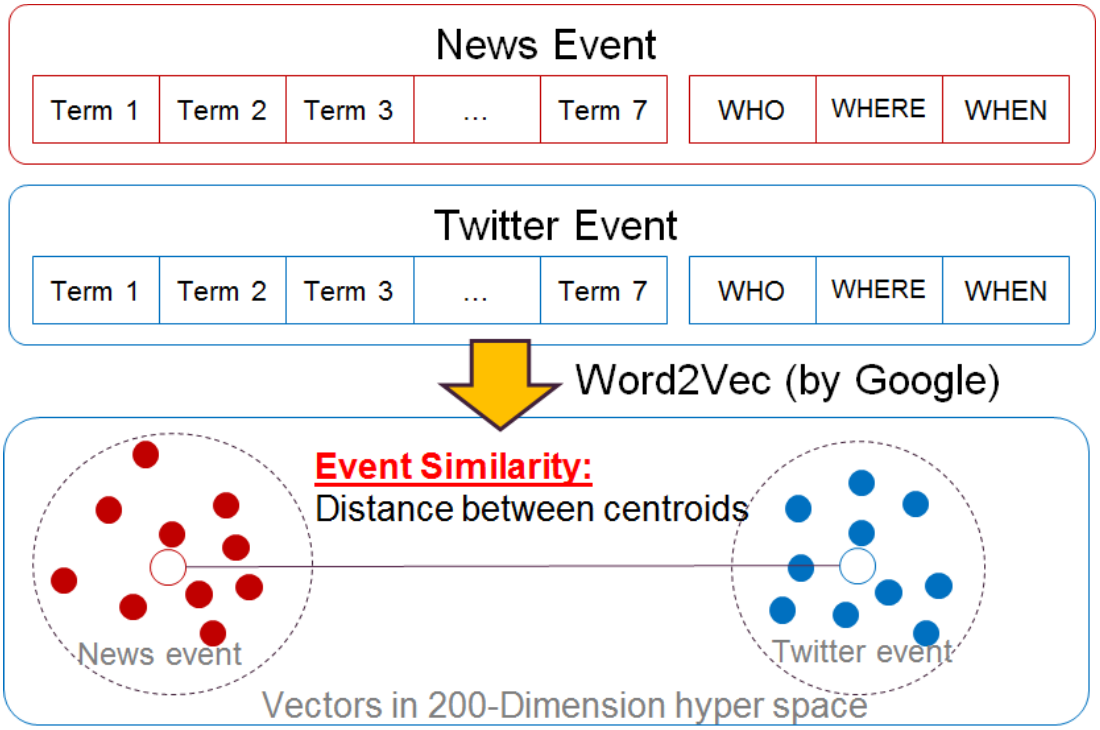

Figure 3. Event Similarity Measured by Word Vectors

As demonstrated by above figure, we explored measuring the event distance by deep learning. A set of vectors are created by using the Word2Vec tool (Deep Learning tool by Google). According to our definition, each event is composed by *topic terms* and *named entities* (WHO, WHERE, WHEN). With Word2Vec, we trained word vectors by taking all the news and all the tweets as corpus. For every word (topic terms or named entities), a vector reflecting its information in the corpus is produced. The dimension of the vectors is 200. In this means, a set of vectors is created for every event, which forms a sphere in the 200-dimension hyperspace. We can measure the distance between two events by gauging the distance between the centroids of these spheres.

At present, the accuracy of the similarity calculation is limited. There are two reasons for that:

- The small corpus.
  In order to create accurate vectors for words, Word2Vec needs a large corpus, usually at least 200M. However, we only have about 30M corpus. According to the accuracy measuring tool by Word2Vec, the vector model we trained has an accuracy only 30%. In comparison, the accuracy of a vector model trained from 800M news is more than 80%.
- The centroid calculation method
  At present, we just calculated the centroids of the spheres by making average to their vectors, which was not so accurate. In the future, we are going to improve it by exploring the Minimum Enclosing Ball approach.

4.4 Experiment

4.4.1 New Event Extractor

From 4084 news articles about “Ukraine Crisis”, we extract 10 events ordered chronologically, as shown below.

```
2014/02/28 - 2014/03/01;
Topic 1: [ukraine, yanukovich, crisis, minister, sign, russian]
2014/03/08 - 2014/03/14;
Topic 2: [crimea, ukraine, russia, minister, referendum, ukrainian, vote]
2014/03/09 - 2014/03/10;
Topic 3: [ukraine, crimea, crisis, putin, russia, minister]
2014/03/12 - 2014/03/13;
Topic 4: [russia, bank, sanctions, ukraine, crisis, crimea]
2014/03/14 - 2014/04/12;
Topic 5: [crimea, ukraine, russian, troops, border]
2014/03/16 - 2014/03/17;
Topic 6: [ukraine, tensions, data, rise, shares, china, stocks]
2014/03/20 - 2014/03/21;
Topic 7: [ukraine, house, imf, u.s, bill, white, aid]
2014/03/23 - 2014/03/24;
```

Topic 8: [ukraine, russia, talks, aid, crisis, sanctions, deal]
2014/03/25 - 2014/04/16;
Topic 9: [gas, ukraine, russian, russia, europe, talks, energy]
2014/03/26 - 2014/03/27;
Topic 10: [ukraine, aid, support, government, talks, house, russian]

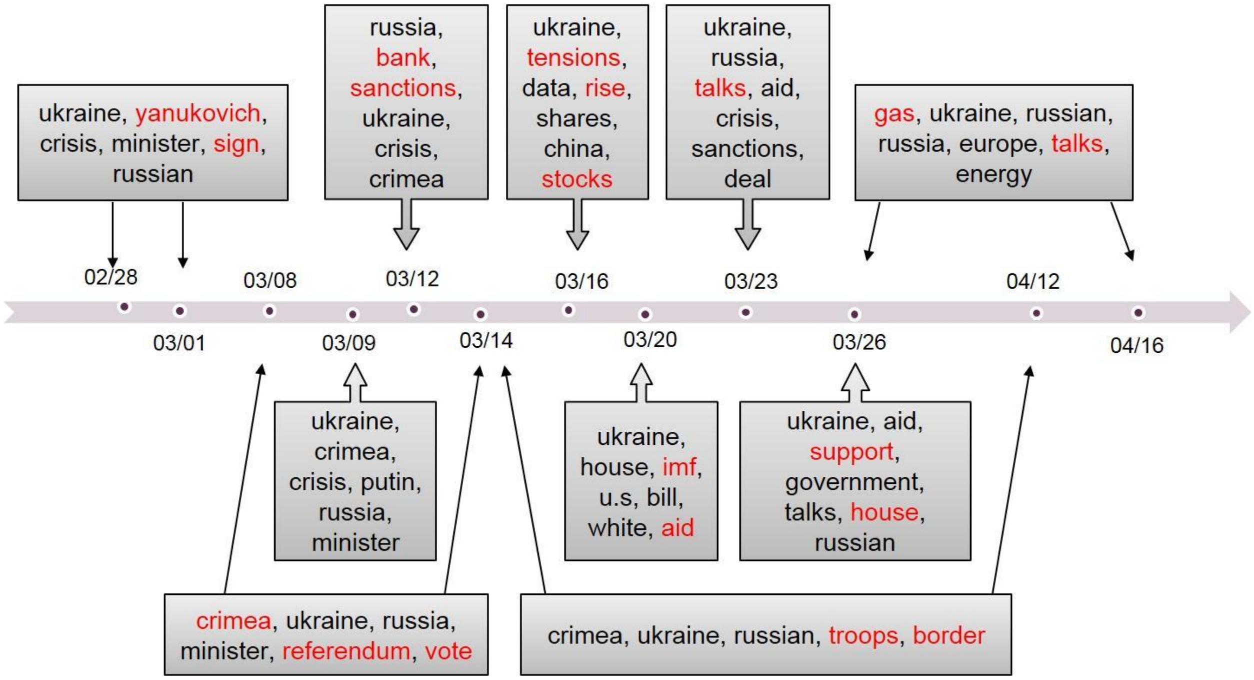
ukraine, yanukovich, crisis, minister, sign, russian
russia, bank, sanctions, ukraine, crisis, crimea
ukraine, tensions, data, rise, shares, china, stocks
ukraine, russia, talks, aid, crisis, sanctions, deal
gas, ukraine, russian, russia, europe, talks, energy
02/28
03/08
03/12
03/16
03/23
04/12
03/01
03/09
03/14
03/20
03/26
04/16
ukraine, crimea, crisis, putin, russia, minister
ukraine, house, imf, u.s, bill, white, aid
ukraine, aid, support, government, talks, house, russian
crimea, ukraine, russia, minister, referendum, vote
crimea, ukraine, russian, troops, border

Figure 4. News Events on a Time Line

There are one or more highly recognizable word(s) in each event topic, which can help users to successfully catch the key words in interested events and distinguish between different topics. And the automatic extraction of temporal information can achieve acceptable accuracy, which basically accordant with the fact.

2014/03/08 - 2014/03/14;
Topic 2: [crimea, ukraine, russia, minister, referendum, ukrainian, vote]

For example, the extracted result shows that Crimea referendum was a very hot news topic during the first half of March. Besides, there was two important events related to this topic occurring around March 8th and 14th. All these can be confirmed by the concrete news facts: 1) Ukraine's acting president annulled the referendum as illegal and unconstitutional on March 7th; 2) a U.N. Security Council draft resolution against Crimea referendum was voted on March 14th.

2014/03/20 - 2014/03/21;

```
Topic 7: [ukraine, house, imf, u.s, bill, white, aid]
```

Topic 7 is another good example, which pinpoints the exact date when a Ukraine-related event involving IMF and U.S. occurred (March 20th). This is corresponding to the news facts that: 1) The International Monetary Fund extended talks with Ukraine on aid package on March 20th; 2) A congressional aide said on March 20th, that the U.S. House of Representatives Foreign Affairs Committee will introduce a bill that provides aid to Ukraine without the IMF reforms.

```
2014/03/25 - 2014/04/16;
Topic 9: [gas, ukraine, russian, russia, europe, talks, energy]
```

The extracted topic about gas issues among Ukraine, Russia and Europe can also demonstrate the validity and reliability of our results. From the output, we can find that March 26th is one of the key point for this topic. In fact, during the Brussels' meetings in that day, Europe and Ukraine both agreed to reduce their dependence on Russian energy supplies. And April 15th is another important day, when Ukraine and Slovakia held talks on reverse gas supplies.

For each event, we provide a key word set to describe this topic, including key time points, followed by the LOCATION and PEOPLE or ORGANIZATION which generally found together with this date. Besides, we also output the top 5 WHO and WHERE under this topic, as a reference. Following is a sample from real output.

```
2014/03/14 - 2014/04/12;
Topic: [crimea, ukraine, russian, troops, border]
WHO: NATO; Oleksander Turchinov; Kerry; Lavrov; Vladimir Putin;
WHERE: Ukraine; Crimea; Russia; U.S.; Kiev;
Combine: [Mar 15 , 2014; Donetsk; Kharkiv; Arbatskaya Strelka; Ukraine; Crimea; Oleksander Turchinov]
Combination: [Mar 29 , 2014; Russia; Ukraine; Crimea; Lavrov; Vladimir Putin]
Combination: [Apr 12 , 2014; Russia; Ukraine; Crimea; Moscow; NATO]
```

From these, we can get some more detailed ideas for the event about the movements of Russian Troops along the Ukrainian border. For example, something happened near Donetsk and Kharkiv round March 15th; and North Atlantic Treaty Organization (NATO) was involved on Apr 12th.

4.4.2 Twitter Event Extractor

We fill our event results of two topics (Topic 1 and Topic 2) in a timeline.

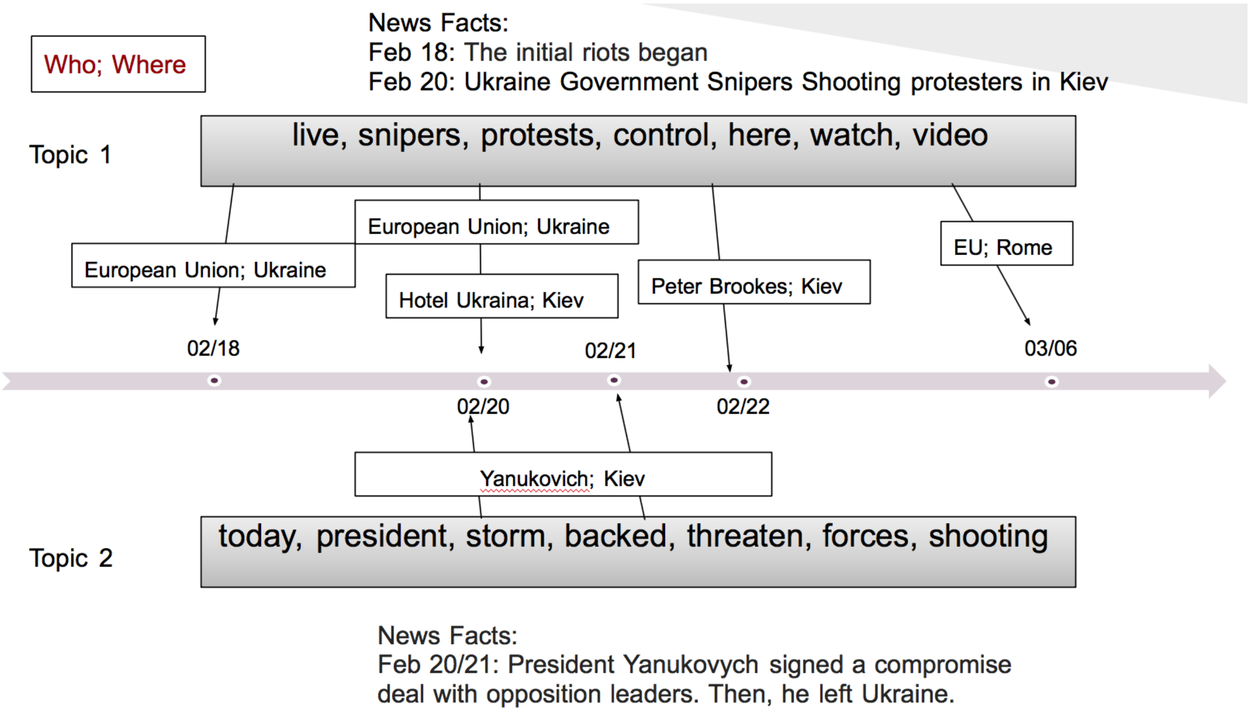
News Facts:
Feb 18: The initial riots began
Feb 20: Ukraine Government Snipers Shooting protesters in Kiev
Who; Where
Topic 1
live, snipers, protests, control, here, watch, video
European Union; Ukraine
European Union; Ukraine
Hotel Ukraina; Kiev
Peter Brookes; Kiev
EU; Rome
02/18
02/21
03/06
02/20
02/22
Yanukovich; Kiev
Topic 2
today, president, storm, backed, threaten, forces, shooting
News Facts:
Feb 20/21: President Yanukovych signed a compromise
deal with opposition leaders. Then, he left Ukraine.

Figure 5. Twitter Events on a Time Line

LDA topic modeling results of 5 Topics:

Topic 1: live, snipers, protests, control, here, watch, video
Topic 2: today, president, storm, backed, threaten, forces, shooting
Topic 3: sector, nazi, gobierno, nato, full, coup, day
Topic 4: east, southeast, vs, send, hoax, separatists, russians
Topic 5: right, gas, ukraine, now, more, orders, minds

Event Extraction Results of 5 Topics with FP Growth:

Topic 1:

```
{Ukraine;Feb 18, 2014;The European Union} 4
{Ukraine;Mar 05, 2014;European Union} 4
{EU;Ukraine;Feb 20, 2014} 3
{Feb 22, 2014;Peter Brookes;Kiev} 2
{Hotel Ukraina;Kiev;Feb 20, 2014} 3
{Ukraine;Feb 19, 2014;The European Union} 2
{EU;Rome;Mar 06, 2014} 2
```

Topic 2:

```
{Kiev;Feb 20, 2014;Yanukovich} 2
{Valencia;Nicosia;Feb 21, 2014} 31
```

Topic 3:

```
{Ukraine;USSR;Feb 27, 2014} 4
{Communities Digital News;Mar 04, 2014;Ukraine;Kiev} 2
{Olesya Zhukovskaya;Kiev;Feb 20, 2014} 7
{Ukraine;BBC;Kiev;Feb 21, 2014} 2
{Ukraine;BBC News;Feb 20, 2014} 2
{Ukraine;Feb 25, 2014;Yanukovych} 3
{Ukraine;Keystone;Feb 20, 2014} 3
{CNBC;Why Crimea;Mar 03, 2014} 2
```

Topic 4:

```
{Earth;Putin;Feb 25, 2014} 8
{Vlad Putin;Adolf Hitler;Mar 01, 2014} 5
{Pshonka;Kliuyev;Feb 26, 2014} 2
```

Topic 5:

```
{Crimea;Mosca;Mar 01, 2014} 2
```

5. Acknowledgements

First of all, we would like to thank the instructor of this CS 6604 course, Dr.Edward Fox, who granted us freedom to adjust the topic of this project to what we are interested in. In addition, Dr.Fox kept offering comments and suggestions during the progress of our project. We also appreciate the help from our client, Mohamed Magdy, who provided data set and suggestions to us. Besides that, we are also grateful to those who have helped this project in different ways, Dr.Weiguo Fan, Dr.Andrea Kavanaugh, and Sunshin Lee.

6. References

[1] Petrovic, Saša, et al. "Can twitter replace newswire for breaking news." *Seventh International AAAI Conference on Weblogs and Social Media*. 2013.
[2] Dou, Wenwen, et al. "Event Detection in Social Media Data." *IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content*. 2012.
[3] Balahur, Alexandra, and Hristo Tanev. "Detecting Event-Related Links and Sentiments from Social Media Texts." *ACL 2013* (2013): 25.
[4] Lobzhanidze, Aleksandre, et al. "Mainstream media vs. social media for trending topic prediction-an experimental study." *Consumer Communications and Networking Conference (CCNC), IEEE, 2013*
[5] The Stanford NLP Group, Stanford CoreNLP package V3.3.0, 2013, http://www-nlp.stanford.edu/software/corenlp.shtml
[6] The Stanford NLP Group, Stanford Named Entity Recognizer package V3.3.0, 2013, http://www-nlp.stanford.edu/software/CRF-NER.shtml
[7] McCallum, Andrew, Mallet package V 2.0, 2013, http://mallet.cs.umass.edu/
[8] Apache community, Apache Lucene Core package V4.6, 2013, http://lucene.apache.org/core/
[9] Mikolov, Tomas, etc, Word2Vec package, 2013, https://code.google.com/p/word2vec/
[10] The Machine Learning Group of the University of Waikato, Weka package V3.6, http://www.cs.waikato.ac.nz/ml/weka/
[11] BigPeng, FP-Growth package, 2013, https://github.com/BigPeng/FPtree
[12] Radim Rehurek and Petr Sojka, Gensim LDA component, http://radimrehurek.com/gensim/index.html
[13] Hoang, Dat, PyNER python library, https://github.com/dat/pyner
[14] Han, Jiawei, Jian Pei, and Yiwen Yin. "Mining frequent patterns without candidate generation." *ACM SIGMOD Record*. Vol. 29. No. 2. ACM, 2000.
[15] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).