Presidential Endorsement Event Extraction

Adeela Huma
Department of Computer Science,
Virginia Tech
ahuma@vt.edu

ABSTRACT

Event Extraction from unstructured data such as news articles is a challenging task because text in news articles is not labelled for person or events and without human intervention its not possible to suggest if article is related to a particular event or not. This study proposes a different approach to event extraction by finding event relationship using machine-learning algorithm without sentence labeling for relations and named entities. The dataset used are the news articles for 2016 presidential endorsement. Political Scientists have long known that endorsements have been among the best predictors of which candidates will succeed and which will fail. If we assign this task to machine and it has to extract endorsement event from newspapers, it may not find any event. This exercise might be waste because news articles might not even belong to presidential endorsement event.

Keywords

Event Extraction, Natural Language Processing, Support vector machine (SVM)

1. INTRODUCTION

With the increasing amount of data and the various number of digital data sources, utilizing extracted information in decision-making processes becomes increasingly urgent and difficult. The problem is that most data is initially unstructured, i.e. the data format loosely implies its meaning and is described using natural, human-understandable language, which makes the data limited in the degree in which it is machine-interpretable.

Event extraction from unstructured data such as news articles could be beneficial for Information Extraction systems in various ways. For instance, being able to determine events could enhance the performance of personalized news systems [19], as news articles can be selected more accurately, based on user preferences and identified topics (or events). Furthermore, events can be useful in risk analysis applications, prediction and decision making support tools.

Event Extraction task includes extracting named entities and relation between those entities. ICEWS [1] and GDELT [2] are two systems that use TABARI [17] (Textual Analysis by Augmented Replacement Instructions) software to extract political events. It uses regular expression to do event encoding. Other techniques used are machine learning based event encoding uses patterns at sentence level. The idea of using information in articles, blogs, social media to forecast important societal events has recently been discovered in several domains, including Google's "flu trends" project [3] and the US IARPA "Open Source Indicators" program [5] and has led to the development of systems like Embers [4].

The United States presidential election of 2016 will be the 58th quadrennial U.S. presidential election and will be held on Tuesday, November 8, 2016. Before any votes are cast, presidential candidates compete for the support of influential members of their party, especially elected officials like U.S. representatives, senators and governors. During the period known as the "invisible primary", these "party elites" seek to coalesce around the candidates they find most acceptable as their party's nominee. Over the past few decades, when these elites have reached a consensus on the best candidate, rank-and-file voters have usually followed.

In presidential primaries, endorsements have been among the best predictors of which candidates will succeed and which will fail. These endorsements can serve several purposes. In some cases, they directly influence voters who trust the judgment of governors and members of Congress for their party. In other cases, endorsements serve as a signal to other party elites. It tells others who is acceptable and who is unacceptable.

Endorsements aren't a foolproof predictor. In 2008, more Democrats initially endorsed Hillary Clinton than Barack Obama (although Obama had some support). Still, a steady flow of endorsements for Obama after his early successes in states like Iowa and South Carolina helped to signal that he was an acceptable choice among party elites and presaged his success in other states.

Predictions of U.S. presidential election results rely on careful measurement and collection of the endorsement behavior from newspapers, web sites of candidates (the U.S. Senate, and the U.S. House of Representatives, Governor, Lieutenant Governor, Attorney General, Secretary of State, etc.), blogs of every candidates and so on. However, it is not very easy to gather all the necessary and relevant data or even have access to latest news due to ones limited access and knowledge. In this paper we approach the objective of extracting endorsements as an event extraction task.

There exists a vast amount of unstructured electronic text on the Web, including news article, Blogs, Tweets and Facebook post and so on. How could a human be assisted to understand all of this data? And after understanding these data how could human extract interested events from the data and predict future events from it. A standard idea is to turn unstructured data into structured data by annotating semantic information of that data. However diversity of data make human annotation difficult.

Current advanced named entities recognizers (NER), such as Stanford NER can automatically label data with high accuracy. However, the whole relation extraction process is not a trivial task. The computer needs to know how to recognize a piece of text having a semantic property of interest in order to make a correct annotation. Thus, extracting semantic relations between entities in natural language text is a crucial step towards natural language understanding applications. In this paper, we will focus on methods of recognizing relations between entities (presidential candidates and endorsers).

For 2016 presidential election, FiveThirtyEight [8] has collected endorsements for presidential candidates by senators, governors and house representatives. Similarly wiki [16] is tracking newspaper endorsements for each presidential candidate. The purpose of this study to take the endorsements data provided by [8][16] and extract endorsement event for candidates.

2. RELATED WORK

In event extraction, ICEWS [1] and GDELT [2] are two noticeable systems that use TBARAI [17] software for political event extraction. TABARI uses a system of pattern recognition to do its coding. Three types of information are used: a) Actors: These are proper nouns that identify the political actors recognized by the system. b) Verbs: Because event data categories are primarily distinguished by the actions that one actor takes toward another, the verb is usually the most important part of a sentence in determining the event code. c) Phrases: Phrases are used to distinguish different meanings of a verb for example PROMISED TO SEND TROOPS versus PROMISED TO CONSIDER PROPOSAL and to provide syntactic information on the location of the source and target within the sentence

TABARI relies on sparse parsing of sentences primarily identifying proper nouns (which may be compound), verbs and direct objects within a verb phrase rather than using full syntactical analysis. This approach requires manually identifying patterns for a specific event and then finding those patterns in all articles.

McClosky et. al.[18] use Dependency parser and simple perceptron to find events but this approach also requires hand labelling of data at sentence level. These techniques are more expensive and require more effort to prepare training set and limit scalability.

Other approaches used are Event extraction as Dependency Parsing [18] that employs use of dependency parser to extract local as well as global events. This approach helps to retrieve desired events as well related events. In our case, this approach can help us find endorsement event as well as the events that led to a particular endorsement.

Almost all approaches requires event labeling at sentence level and there could be many ways to express the same event in natural languages and finding out all possible patterns can be cumbersome process.

3. PROBLEM DEFINATION

Event Extraction requires identifying sentence patterns for relations and using patterns to find endorsements. Labelling sentences and identifying patterns manually is cumbersome process. *This study proposes to use machine-learning algorithm to find relations by classifying events at document level and then find entities.* With this approach classifier will separate the endorsement documents from non-endorsements. By using this approach we do not need to find sentence patterns related to endorsement event.

4. APPROACH

The proposed approach is to first separate endorsement articles from non-endorsement articles i.e. find endorsement relation. Then it runs named entity recognition to find entities in endorsement newspaper. For the 2016 US presidential elections, we would extract endorsement event for each of the presidential candidate. Actors include governors, U.S. senators, U.S. representatives, and newspapers. For relation extraction traditional approaches require pattern identifying at sentence level and it is proposed to find relation in news articles by using classification at document level.

Endorsement event extraction system consists of following major parts:

- Data Scraping
- Relation Extraction
- Entity Extractor

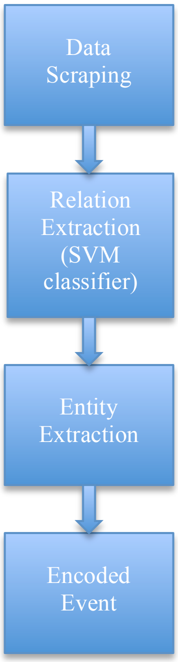

Figure-1 System Architecture

Data Scraping extracts data from many sources:

a) FiveThirthyEight [8] website that provides data about which candidate is endorsed by whom on which date.
b) Newspaper based endorsements for candidates [16]
c) Non-endorsement Political documents for 2016 presidential election [9,10,11,12]
d) Senators, governors, house representatives (actors list) [13,14,15]

The information extracted for endorsements is saved as who endorsed whom on which date.

Endorser	Candidate	Date	ReferenceArticleName	ArticleText

In extraction of endorsement event from news articles, named entities of interest are persons i.e. presidential candidates, senators, governors, house representatives. The list of person names is scraped from wiki and is saved as person_entity dictionary. The newspaper names are also scraped and saved as part of dictionary.

After getting the endorsement and non-endorsement documents. SVM classifier is trained to identify endorsement relation/event.

Once we find that a given article is about presidential endorsement, program finds the entities mentions in it and matches it with dictionary.

4.1 Data Scraping

Data for experiments are scraped from different websites to get presidential endorsement, non-endorsement articles, actors i.e. newspaper names, senators, governors, house representative's names. From FiveThirthyEight [8] website endorsement related data is collected. FiveThirthyEight site has been collecting all the endorsement related data from News articles, Facebook, blogs and twitters. This site has running presidential candidate related article as well as dropped out presidential candidate related articles too.

Web scraper program, collects all the endorsement related news articles. Non-endorsement articles are chosen from news websites about presidential election. Data Scrapper program also get names of newspapers, senators, actors and house representatives to form actors dictionary. This component is written using python requests, BeautifulSoup and Goose libraries [6,7].

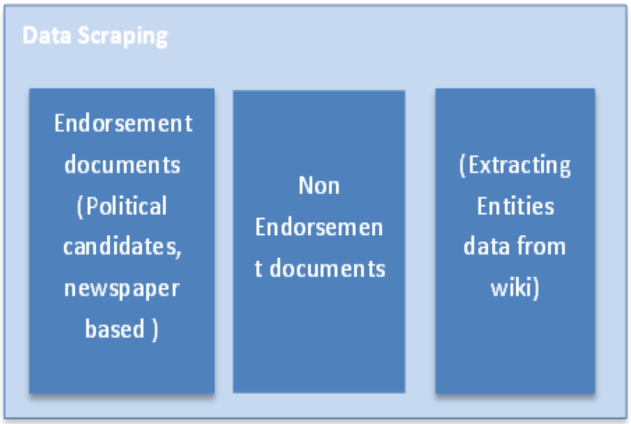
Data Scraping
Endorsement documents (Political candidates, newspaper based)
Non Endorsement documents
(Extracting Entities data from wiki)

Figure-2 Data Scraping Component

Scarping news articles from web has some issues as following:

- The endorsement event was described as an image file.
- The endorsement event was mentioned in a video etc.

Since no text is present in such cases. These articles were ignored during scarping process.

4.2 Relation Extraction (Classifying Endorsement Documents)

The system uses Document level classification to identify if the endorsement relation/event is in a document or not. Input to the system is endorsement and non-endorsement documents. Features are extracted using **'it-idf'** approach. Text processing is done using tm (text mining) package in R- language. This process includes:

- deleting preamble text in each of the documents
- converting all text to lower case
- stop word removal
- punctuation removal
- removing whitespaces

Since text in news articles consists of a lot of words that do not contribute in identifying endorsement event such as stop words etc., it's preferable to delete them. After pre-processing the remaining text is converted to Document Term Matrix (DTM).

A **document-term matrix** or **term-document matrix** is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. In a document-term matrix, rows correspond to documents in the collection and columns correspond to terms. There are various schemes for determining the value that each entry in the matrix should take e.g. term-frequency, tf-idf etc.

To computer the terms in Document Term matrix, **tf-idf** [20,21,22] approach is used. tf–idf, short for **term frequency–inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus/documents. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

Typically, the tf-idf weight is composed by two terms:

- the first computes the normalized Term Frequency (TF), aka. the number of times a word appears in a document, divided by the total number of words in that document;

- the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

In text classification domain each word is a feature: whether it is present in the document or not or how many times it appears. After pre-processing, the resulting Document Term Matrix has 13661 terms (words). Most of the matrix has sparse terms. The sparse terms is not a good choice for feature, it merely adds noise. After removing 80% of the sparse terms from the matrix, we get 84 terms. Following is the word cloud and frequency plot of terms obtained after removing sparse terms.

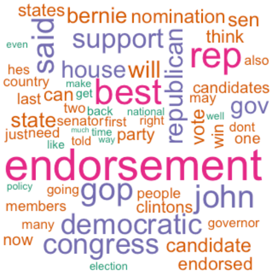

Figure 3- Word cloud of Endorsement Documents

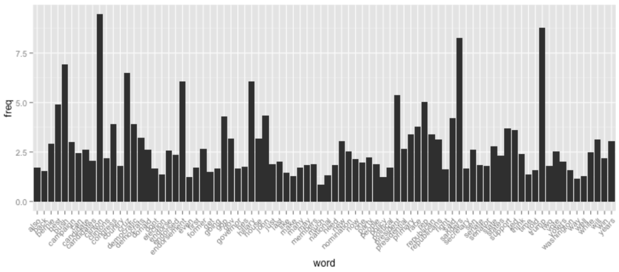

Figure 4- Frequency Plot of words

The Terms obtained after removing sparse terms can be used as features to classify endorsement documents from non-endorsement containing news articles

Support Vector machines (**SVM**) is used for binary classification of news articles using linear kernel function. Linear kernel is suggested for text classification. For binary classification, system uses two class labels:

- Endorsement (1)
- Non-endorsement (-1)

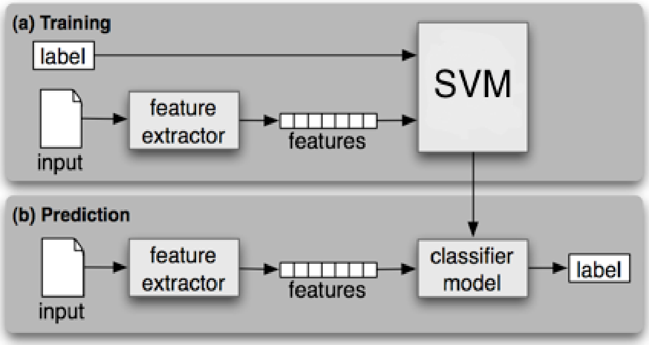

Figure 5: Relation Extraction Component

Before data processing each article is assigned a class label. Using tm module of R-language we get Document term matrix and then DTM is converted to vectors.

For classification SVM is trained on 70% data and remaining 30% is used for testing. Training is done on 70% of randomly chosen vectors using "linear" kernel with cost =1. Following are the result of experiment:

Confusion Matrix: A confusion matrix is a table where each cell [i,j] indicates how often label j was predicted when the correct label was i.

	endorsement	non-endorsement
endorsement	104	12
non-endorsement	14	29

SVM model's accuracy using linear kernel is: **83.6478%**. After this process we have identified the news articles that would have 'endorsement relationship'.

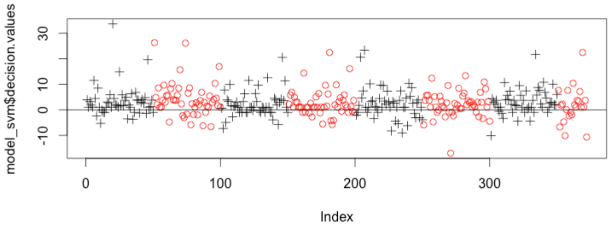

Figure -6 SVM Prediction Results

Figure 6 shows SVM's output. X-axis represents documents; Y-axis represents decision values (1 to -1). '+' symbol shows the support vectors and circles shows the class labeled data.

4.3 Entity Extraction

After SVM classification, we get endorsement news articles and next step involves finding person entities from news articles that in this case are names of newspaper, senators, governors and house representatives. Once person entities are found they are matched with actors dictionary. If there is a match these are taken as entities of endorsement event.

Named Entity Recognition (NER) is a specific method of the Information Extraction (IE) task dedicated to classifying phrases in text that refer to entities like people, date, organizations, locations, expression of times, percentage, monetary values and currency amounts and extracting their semantics. NER is also known as entity identification, entity extraction or entity chunking. NER tasks can be broken down into two sub tasks: a) identifying the boundaries of the NE, and b) identifying its type.

This component is written using NLP library in R- language.

This component first finds the sentence level tokens, annotates words and then finds person entities. This component finds correct entities for articles that are short and direct like below:

> **"Sam Graves** endorses **Ted Cruz"**

System finds correct entities. Unfortunately all news articles are not that concise and do not talk about event in a direct way and also uses comparison with other candidates to justify their choice. Due to this problem this component of architecture find multiple entities. Details are defined in next section.

4.3.1 Entity Extraction Issues

All news articles are not that concise and do not talk about events in a direct way and also uses comparison with other candidates to justify their choice. Due to this problem this component of architecture find multiple entities. Below are details of such problems:

1. Since system runs named entity extraction on whole article, more than one entity are found. For example in following chunk of text from endorsement for Hilary Clinton, other people were mentioned for comparison. But they are not part of endorsement relation. System list multiple entities in this case.

> **"…. Warren** maintains that she isnt running for president, but that didnt stop the grassroots group MoveOn.org from calling for her to run on Sunday within hours of Clintons debut as a candidate. One of **Warrens** former speechwriters and campaign consultants, **Marla Romash**, is a longtime confidante of **DeLauro** and a former press aide to **Chris Dodd** and **Al Gore**. …."

2. If whole article is in Capital letters then it is hard to identify Person names.

3. If some sentences have state name followed by actors than this component takes state name plus actor name as a Person.

4. A Person's name can be written in many ways such as first name followed by last name or vice versa. In this case same person's name is identified as separate entities by NER component. In the below example NER component found "Hillary Clinton" and "Clinton" as separate entities.

> "…To no ones surprise, U.S. Sen. Richard Blumenthal tonight formally endorsed **Hillary Clinton** for president. She knows how to get things done, he told a Ready for **Hillary PAC** fundraiser at the Westport home of Ann Sheffer and Bill Scheffler…"

These can be avoided by not running named entity recognition on whole article. Instead run it on main paragraph as TABARI does.

5. CONCLUSION

This study shows that we can identify event relation by classifying news articles at document level without annotating patterns for relations at sentence level. For endorsement event-finding relation at document level it gives 83.6% of accuracy. This approach can be extended to other events of interests. Only problem is system extracts multiple entities from an article. It can be resolved by using only main paragraph that gives summary of article.

6. References

[1] S. P. O'Brien. Crisis Early Warning and Decision Support: Contemporary Approaches and Thoughts on Future Research. Int. Stud. Rev., 12(1), Mar. 2010.

[2] K. Leetaru and P. Schrodt. GDELT: Global data on events, location, and tone. ISA Annual Convention, 2013.

[3] S. Cook, C. Conrad, A. L. Fowlkes, and M. H. Mohebbi,"Assessing Google flu trends performance in the United States during the 2009 Influenza virus A (H1N1) pandemic," PLoS one, vol. 6, no. 8, 2011.

[4] N. Ramakrishnan, P. Butler, S. Muthiah, N. Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz, et al. 'beating the news' with embers: Forecasting civil unrest using open source indicators. arXiv preprint arXiv:1402.7035, 2014.

[5] http://www. iarpa.gov/index.php/research-programs/osi

[6] https://www.crummy.com/software/BeautifulSoup/

[7] https://github.com/grangier/python-goose

[8] http://projects.fivethirtyeight.com/2016-endorsement-primary/#endorsements

[9] http://www.huffingtonpost.com/news/elections-2016/

[10] http://www.breitbart.com/2016-presidential-race/

[11] http://www.nytimes.com/pages/politics/index.html

[12] http://www.cnn.com/specials/politics/2016-election'

[13] http://www.house.gov/representatives/

[14] https://en.wikipedia.org/wiki/List_of_current_United_States_governors

[15] https://en.wikipedia.org/wiki/List_of_current_United_States_Senators

[16] https://en.wikipedia.org/wiki/Newspaper_endorsements_in_the_United_States_presidential_primaries,_2016

[17] http://eventdata.parusanalytics.com/tabari.dir/TABARI.0.8.4b3.manual.pdf

[18] D. McClosky et al., Event Extraction as Dependency Parsing, ACL 2011

[19] Borsje, J.,Hogenboom,F.,Frasincar,F.:Semi-AutomaticFinancialEventsDiscovery Based on Lexico-Semantic Patterns. International Journal of Web Engineering and Technology 6(2), 115–140 (2010)

[20] K. Sparck Jones. "A statistical interpretation of term specificity and its application in retrieval". Journal of Documentation, 28 (1). 1972.

[21] G. Salton and C. Buckley. "Term-weighting approaches in automatic text retrieval". Information Processing & Management, 24 (5). 1988.

[22] H.WuandR.LukandK.WongandK.Kwok."InterpretingTF-IDF term weights as making relevance decisions". ACM Transactions on Information Systems, 26 (3). 2008.