ONTOLOGY USE IN SUMMARIZATION FOR THE TWITTER ANALYSIS ON US ELECTIONS

Authors:

Vilas Mamidyala UMKC MS(CS)
vm3k5@mail.umkc.edu

Vikesh Padarthi UMKC MS(CS)
vpn7d@mail.umkc.edu

Ranjithreddy Bhumireddy UMKC MS(CS)
rbpr4@mail.umkc.edu

Dinesh kumar Bandam UMKC MS(CS)
dbk7f@mail.umkc.edu

Abstract

This article is completely about the prognosis of US elections 2016. Although we have tweets from twitter, it is not possible to come to know what exactly those tweets talk about. Thus we worked on analysis of tweets and applied ontology to these collected tweets in order to get the knowledge graphs for the summarized tweet text output. This analysis is now about applying NLP processing workflow and generating the ontology. We had further applied clustering and naïve bayes algorithm to it. Further in the paper we will discuss the detailed steps towards summarization.

1 Introduction

We know that the whole world is awaiting to hear the result of US election which are going to be released by the end of this year. Everyone would like to see how these elections are going to be held. One has an anxiety that who is going to win and what actually the people opinion is and who has more probability to win. These questions stimulate our work towards collecting data about politics which clears all our skeptic things about elections. Since many of the things related to students and their future who have more excitement and worry to get to know the result. Our main motivation behind this project is to analyze the data present in social media like twitter and plot some graphs which shows about which candidate is more famous in social media, the probability of who will be getting elected.

Main objective of this project is to use NLP, machine learning knowledge to predict the outcome of election result. Using these we can summarize the result of various blogs, news, and editorial matters in newspapers which are related to elections. We will first plot some graphs based on the twitter data which we have collected and we want to analyze various text data present in the World Wide Web like Wikipedia and summarize these papers.

As explained earlier by performing these operations using NLP, Machine Learning we want to predict the outcome of the US elections and various views about US elections by the people around the world. The output will be ontology graphs which are developed by analyzing the data sets which are related to US elections.

2 Related work

The paper “An Ontology-based Approach to Text Summarization” is one approach which tells us and focusses on the importance of summarization. The approach they followed

is to use of hierarchical ontology which aims at improving the sentence semantics. They have used a SVM classifier to identify the summary sentences. This approach gave us input for our project approach where we followed a similar method of implementation which uses ontology generation and then applying clustering to it.

3 Proposed solution

Architecture:

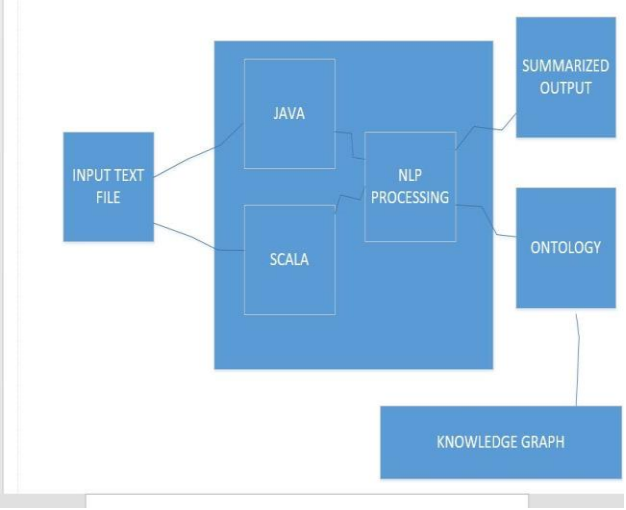

Fig 3.1 Architecture

First of all, we will collect tweets which is done by using java code and then this is applied with nlp coding written in java which uses Scala platform.

NLP processing has been performed to this java code and the result is summarized output from which we can get ontology graph and knowledge graph.

Here as shown in the architecture we have ontology and summarized output as result of our code written in Scala. Scala is having implementation of naïve bayes algorithm and used feature vector for generating the ontology.

Workflow:

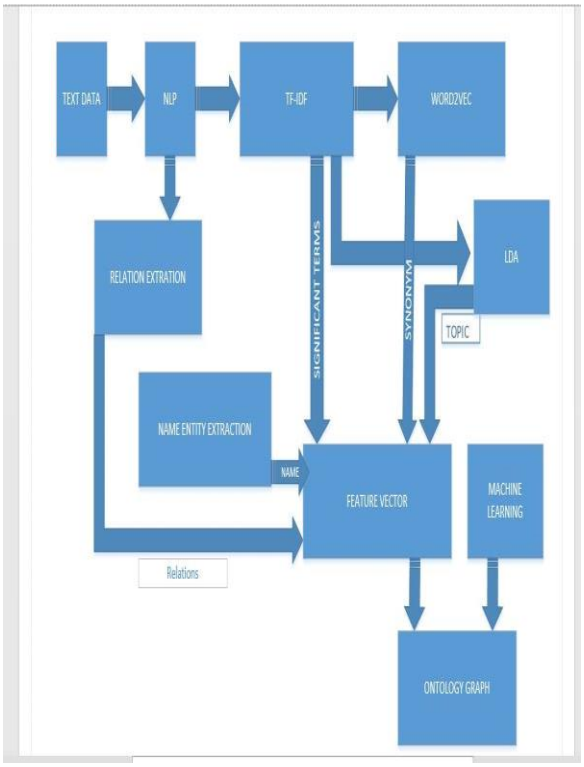

Fig 3.2 Workflow diagram

An NLP processing has performed to the given input data i.e data from tweets, it is given to the TF-IDF operation followed by word2vec operation LDA. From the result of NLP, a Relation Extraction has been done which is given to the feature vector along with the Name Entity Extraction result. This Feature Vector takes also TF-IDF result and word2vec result including LDA result.

Along with the help of Machine Learning Feature vector generates an Ontology Graph.

4 Implementation

As part of implementation we had started first by collecting tweets. This is done by two as mentioned bellow: We have collected twitter data:

- Using CURL command in LINUX terminal by filtering the keywords such as Hillary Clinton, Trump, US elections etc…
- Using java code which extracts only tweets information o same keywords mentioned earlier.

Stages Followed:

- Implemented word count program using Scala:

Initially, we have collected data from the tweets. We worked on not only on the words in a tweet but also how many tweets are collected at a time and word count is performed on all the data and then an output has been generated which results number of words in the complete file. Finally, an average of word count has calculated for all the tweets. This code is written scala language which gives as output as each word and their occurrence.

- Implemented NLP program:

NLP is a part of Artificial Intelligence. This Redmond-based NLP group mainly develops a type of algorithms that provides a processing of texts in an efficient manner and to use this in computer applications in an easy manner. We can say that the capacity of a system for speech recognition as an outcome of a human voice can be the perfect example of NLP. Our program which is written for NLP will do the basic steps such as Lemmatization, POS tagging and then NER.

Lemmatization: Getting the representation of root word or dictionary word for the words in a tweet.

POS tagging: It assigns the parts of speech for the words in the tweets those shows whether it is a noun, verb or adjective.

NER: Allocating a label to each and every word and then identifies to which group it relates to like whether the word is in correspondence with a person if the word is he/she.

- Implemented TF-IDF:

Term frequency (TF) – Inverse document frequency (TF-IDF): This is a statistical approach which finds the words that are important from a given text collection. By using the stand ford university given codes, we had made changes which suits our topic and then we had implemented the TF-IDF calculation.

A simple example of NLP is the capacity of a system to recognize human speech as what it is spoken.

- Implemented Word2vec:

  word2vec is nothing but representing words in the form of vectors. Image and audio processing systems work with significant, large-dimensional data files that are encoded as vectors of each raw pixel-intensities for source image or graphic data, or i.e. we can also say for those the PSD coefficients for voice data. For those tasks such as object or voice recognition we know that all the collected data need to be efficiently does the task which is encoded in the collected file output (since humans can do these type of tasks from the unprocessed data). However, NLP systems earlier treat words as discrete atomic symbols, and thus for example a word or a name ‘cat' was called as Id537 and similarly 'dog' as Id143. These provided encodings are constant, and results in no helpful data to the system for the companionships that might between the singular symbols.

- Implemented WordNet:

  WordNet is a huge lexical database of English language. In this, we actually group nouns, verbs etc as sets of cognitive synonyms each displaying a unique concept. we have given words that are related to our project and then it showed their synonyms and related words.

- Implemented NER:

  In this process we had taken the political words related to USA and then we have given a label of "POLL" to them and then trained the model. Once it is done, we have then tested our model by giving some political tweets and which will identify the political word for the given tweets file.

- Implemented Feature vector generation:

  Future vector generation involves some steps where we have to apply different logical steps and methodology. As part of this, first we have to develop data sets which are called RDD (Resilient Distributed Dataset). This step involves only making large text into different data sets and so that the next each operation can be performed on these small data sets instead of making on large data text files. We apply Lemmatization on these data sets and we can observe from this that it gives us the words which are present in dictionary like if we have mangoes in the text file after lemmatization it gives us mango. Next we will generate data frames from these data sets. After generating these data frames, we will apply tokenization on these data frames. We will apply stop word remover and NGram on this and

finally we will generate word2vec. Feature vector generation involves all these steps.

- Generated ontology and used protégé tool for its visualization:

  Ontology is nothing but creation of ontology graphs. As part of this we will generate graphs where unstructured data is developed into structured data. Suppose if we know that a novel is written by some author and if it is written in text it will be hard to understand and there will be grammar mistakes some times. If it is represented in the form of graphs it will be easy to read quickly and the relation can be represented on the graph. As part of the project, we have developed ontology graph where it shows about elections in US. In which year the elections are running and who are the candidates and from which parties they are. So, in this way everything can be represented in one simple graph where the whole text page can be represented. In this way summarization can be found easily in graphs.

```
Corpus summary:
    Training set size: 2 documents
    Vocabulary size: 13 terms
    Training set size: 15 tokens
    Preprocessing time: 0.558577732 sec

Finished training LDA model.  Summary:
    Training time: 4.352653153 sec
    Training data average log likelihood: -20.115197054435626

3 topics:
president win usa election state currently majority female clinton lead oppostion trump virtue
usa win president virtue trump oppostion lead majority female currently clinton election state
usa win president clinton state lead currently election female majority virtue trump oppostion
0.0
0.0
0.0
```

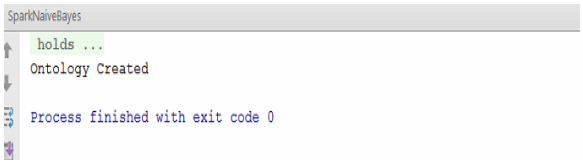

Fig:4.1 Output Ontology Creation

- Protégé tool:

  Here we have used Protégé tool to view the onology created. The below figure shows us how the ontology looks like.

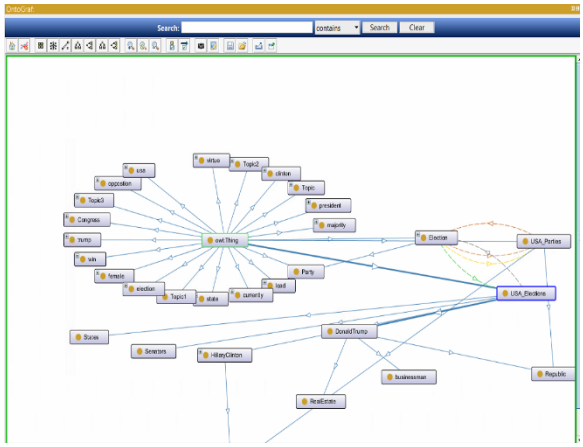

Fig 4.2 Ontograf

- Clustering:

  Here, in this process we use Naive Bayes algorithm. Naive Bayes algorithm is a technique based on the Bayes Theorem. It is a classifier, that assume a feature of a class exist is not related to existence of any other feature of it. Here we had done one important step of our project that is the classification of tweets to a positive and negative clusters. This has been done by using the trained

data set. We had first trained our model by giving 50 positive words and collected some tweets.

Similarly, for the negative words. We had then trained the model by giving these positive and negative tweets. Now our system is well trained to classify both positive and negative clusters. Now we had further continued and tested our model by giving the tweets data that we had collected earlier. This will now classify our input file based on the trained set. This program will give the result as 1.0 or a 0.0 as output. 1.0 indicates that our input is negative and 0.0 indicates a positive tweets file. when positive 50 negative 30 tweets are given as input, we can see that output is 0.0 which tells us that here positive tweets are more in number and it is domination the negative tweets. When negative 50 positive 30 is given as input we can see that output is 1.0 which tells us that here positive tweets are more in number and it is domination with the negative tweets.

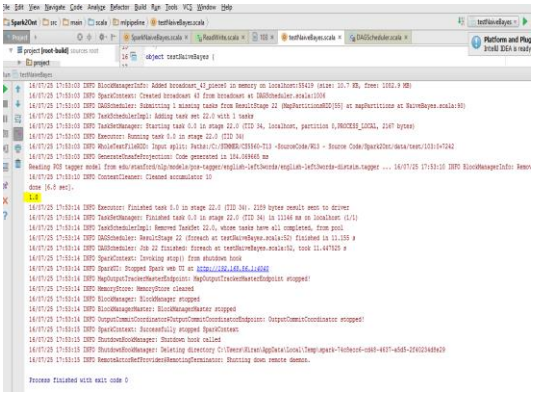

Fig 4.3 Clustering testing output

- API services are implemented for summarization:
  This is our main motto of our project to summarize the given text files which contains the tweets. So here we had used an API service to this for our project. Here we had written three different codes for the summarization. First one is applying summarization for a given URL. This URL can be any URL and this code will now summarize the content for the given web site. In our experiments we had given the url like twitter.com/hillaryclinton which gives the summary of tweets that were tweeted by Hillary. Second is applying this technique on the given string raw tweet data. This will now summarize the raw tweets that are given as input. Third code in this approach is to summarize for the given text file. Here we will give our actual tweet file as input and this will summarize the given input.

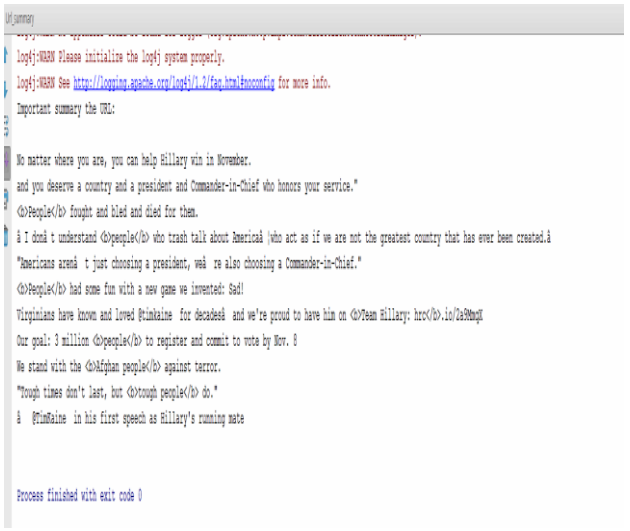

Fig 4.4 Output of Summarization for a URL

Project Related URLs:

Source code:
https://github.com/vilasmamidyala/KDM_SM16_SM

You tube demo url:
https://www.youtube.com/watch?v=StuOZs9IkwU

5 Results and Evaluation

- **Accuracy:**
  For TF-IDF Feature vector is 0.2857 with respect to our input tweets, whereas for simple feature vector we achieved an accuracy of 0.5(Figure 5.1).

  Here when we ran the Naïve Bayes algorithm we got the above feature vector and the confusion matrix. We could also observe that it has given the accuracy of 0.5 as well.

  When we performed clustering for the tweets with positive and negative we had achieved 100% accuracy for the classification. This is a good achievement with respect to clustering.

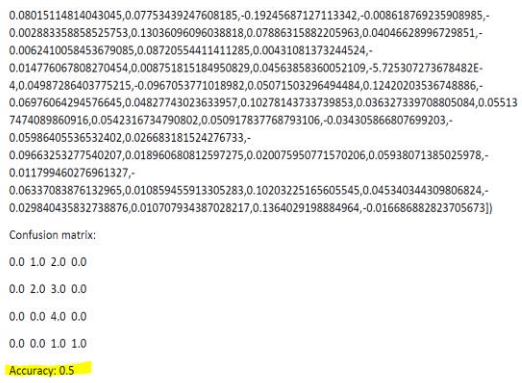

Fig 5.1 Accuracy

- **Run Time Performance:**

  Ontology Run Time: 15 min

  Summarization:

  - For a given file: 1minute
  - For a given URL: 1minute
  - For a given text: 1minute

  I. Clustering of tweets with respect to positive and negative words using Naïve Bayes Algorithm: 7 min for training and 3 min for testing.

  II. OWL file opening through Protégé: 3 min for viewing Ontograph.

  III. SparQL query execution: 1minute. We have write some basic queries in SparQl which were not shown here in the paper can be found from the github.

6 Conclusion

It is already known that; USA elections are going to play a major role which plays not only in US but also its impact can also see in other countries too. So as a part of this project, we have taken US elections as our domain and performed functions like expecting who will be the president and on a particular situation or any given scenario which candidate is going to stand on positive side and negative side, also summarizing the speeches given by these candidates that are taken from Wikipedia and other sources of news.

7 Future Work/Issues

Issues faced:

1) For small amount of data given as input for NLP processing and for other code executions. We found that these programs are working well and giving better results. The issue has occurred when we had tried implement NLP operation on large amount of data the programs were not able to run properly.
2) We considered taking Twitter data for the first phase. But we want to know whether twitter data can be useful for summarization? Because each tweet will be independent of the other tweets most of the times. This data alone might not help us for summarization. we think we need to take other different source s of data as well. we will try to figure out about what are the other sources that can be included.
We have implemented both summarization and clustering differently. As the issue was we could not able to rate which are important to us and which are not. So that made the difficulty over this research. It could be implemented in other approaches.

Future Work:

We have implemented both summarization and clustering differently. As a part of future work we can implement both the functionalities in a single program. Further extension of clustering can be taken as input for the summarization and we can make summarization of the tweets that are been clustered together.

Implementation of a dynamic web application which can take an input of a text file and gives an output on the web page itself.

8 References

- NLP: http://nlp.stanford.edu/
- SparQL: https://www.w3.org/TR/rdf-sparql-query/
- Ontology: http://homes.cs.washington.edu/~pedrod/papers/hois.pdf
- Protégé: http://protege.stanford.edu/
- NaiveBayes Algorithm: http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab4-NaiveBayes.pdf
- Summarization**:** https://www.semanticscholar.org/paper/An-Ontology-Based-Approach-to-Text-Summarization-Hennig-Umbrath/ab138bc53af41bfc5f1a5b2ce5ab4f11973e50aa
- API :http://www.intellexer.com/