ASPECT BASED SENTIMENT ANALYSIS FROM NEWS ARTICLES

A Project Report

Submitted by

Suhas K S	**201405524**
Shwetha G	**201405525**
Mrugani Kurtadikar	**201405621**

Under the guidance of

Manish Shrivastava

IIIT Hyderabad

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

May, 2016

Abstract

There are wide variety of newspapers that share political, sports, etc news with the readers. However, there is no mechanism to compare the news sentiment from various newspapers. The motivation behind this project is to analyze similar kind of news from different newspapers like The Hindu, FirstPost, Indian Express and provide a comparative analysis of the sentiment from news. The sentiment can be of different opinion holders on several issues present in the news articles. We have applied different techniques like named entity recognition, dependency parsing and coreference resolution to extract speakers and their sentiments from the news. We also used Alchemy API to extract entities and their sentiments.

Contents

List of Figures

Chapter 1

Introduction

1.1 Sentiment Analysis

Sentiment Analysis is the process of deciding whether the given text is positive, negative or neutral. It is also referred to as Opinion mining. Sentiment is the attitude, opinion or feeling toward something, such as a person, organization, product or location [6]. There are different ways to perform sentiment analysis using different Machine learning algorithms, feature extraction from text. A lot of work has been done in sentiment analysis in Twitter. Little work has been done in sentiment analysis from news articles due to :

- lack of available structured data from news articles
- lack of available tagged news data

News articles contain lot of data including discussions, arguments, opinions or statement on some topics. A news article can be mixture of several topics.

News articles may contain opinion holders, targets and topics[8]. Opinion holders can be named entities from Person or Organization category. Some prior work has been done on this. In [8], novel technique using semantic labelling has been followed to extract opinion holders and topics expressed in news media text.

In [9], work has been proposed that assigns scores indicating positive or negative opinion to each distinct entity in the text corpus. This system consists of a sentiment identification phase, which associates expressed opinions with each relevant entity.

There is some work in temporal sentiment analysis. According to [10], Temporal sentiment analysis that analyzes temporal trends of sentiments and topics from a text archive that has timestamps is proposed. The output is graphical with two graph topic graph and sentiment graph.

1.2 Problem Statement

The basic aim of this project is to provide a comparative analysis of opinions expressed by different entities in different news articles. For our experiments we have crawled common news topics like:

1. JNU issue
2. Beef ban
3. Intolerance India
4. Awardwapsi
5. Nirbhaya
6. Net neutrality
7. panama paper leak
8. D K Ravi
9. Salman Verdict
10. Sunanda Pushkar

from The Hindu, FirstPost and Logical Indian sites. We have developed an interface which accepts two entities from users and displays the sentiment score from source entity to target entity. The sentiment score is displayed from all news sources in which it is found. The output contains following:

1. Entity Relation graph - This is a directed graph from subject entity to object entity. It indicates the opinion sentiment from subject to object.
2. Entity Entity graph - This is an undirected graph which indicates cluster of entities from an article. There can be different such clusters obtained from a news corpus.

We shall discuss this in detail in section 4. In order to provide comparative analysis we have chosen controversial issues.

Chapter 2

Approach

We have applied different approaches and later came up with a final solution.

2.1 Algorithm 1

The first approach that we followed was a naive approach. This was a basic step towards sentiment analysis of news data. The basic assumption that we made in this was that news articles contain several sentences. Each sentence is parsed and named entities are extracted from the sentence. The polarity of the sentence is computed using [5]. The polarity score is associated between every two entities obtained from the sentence. Thus we obtained a graph containing all the entities extracted from news corpus. The graph may be connected or it could have multiple clusters depending on the news corpus. If the data is higly topical i.e. on the same topic, we get a single big connected network of entities. If the data is a mixture of several topics, we get multiple clusters of entities.

Algorithm 1 Naive algorithm

```
for each sentence in text do
    score = polarity(sentence)
    entities = namedEntities(sentence)
    Associate score with each entity.
end for
```

2.1.1 Issues with Naive Approach

Though this was a basic step towards sentiment analysis, there were several issues such as:

1. Unable to identify subject and object
2. Polarity score is the overall polarity of a sentence, each entity can have different sentiment on news issue.
3. Coreferences are not resolved.

4. Failed to identify aspects from the articles.

2.2 Algorithm 2

This is an improvised version of naive approach. In this we applied certain techniques to overcome the above issues. Firstly, we used nltk coreference resolution package to resolve reference to same object in the article. Second, used dependency parser to extract subject, object from each sentence. Then identify action taking place between subject and object and associate polarity with the action. In this we used the expression from verb actions to guess the sentiment of sentence. Words like criticized, claimed etc indicate negative sentiment. In this we captured sentiment from sentence as well as verb actions.

The various components used in this approach are :

1. Coreference Resolution: It is the task of finding all expressions that refer to the same entity in a text[1]. This is important in our approach because we are parsing text sentence wise. An entity may be referred by pronouns in further sentences, hence it was required to resolve coreferences from an article. We have used core nltk package for resolving coreferences.
2. Dependency Parser: The dependency parser provide a representation of grammatical relations between words in a sentence. Stanford dependencies are triplets: name of the relation, governor and dependent[2]. We have used dependency parser to extract subject, object from the sentence as well the action between subject and object. We have taken a list of standard verb actions and their polarity. The sentiment score from subject to object is determined by the polarity of the verb action.

Algorithm 2 Algorithm 2

```
for each sentence in text do
    Resolve coreferences
    score = polarity(sentence)
    subject, object = dependencyParser(sentence)
    subject–object = score
end for
```

2.2.1 Issues with Algorithm 2

The issues in this approach are mainly with parser, difficulty in getting the correct subject, object from sentence. It is difficult to interpret the output of parser, It works well for small sentences however, results are not upto the mark for larger sentences. Also, the news data is not well structured.

2.3 Algorithm 3

Alchemy API is capable of identifying Subject-Action-Object relations within HTML, text, or web-based content. It uses sophisticated statistical algorithms and natural language processing technology to analyze information, extracting the semantic richness embedded within.

In this, we have used alchemyAPI developed by IBM [7]. The API can be used over structured as well as unstructered news data. It resolves coreferences, extracts subject and object entities, maps with named entitities and gives sentiment score from subject entity to named entity. For the sentence, "Ugly Bob attacked beautiful Ana.", the output of API is:

Let us see the meaning of each item from the json. From the sentence, first the subject, object and verb are extracted. For the given sentence, subject - Ugly Bob object - beautiful Ana verb action - attacked

The relations array is created in json for every relation found in the sentence. Each relation has a sentiment score for that entity. Subject and object has one more field, which indicates the named entity present in the subject and object.

Object relation contains sentimentFromSubject as one of the field, this field indicates the sentiment of subject on the object entity, which is of our interest. Thus we obtain a similar structure using this as we had obtained using Algorithm 2.

Listing 2.1 JSON example

```
{
    "relations":[
        "subject":{
            "text":"Ugly Bob",
            "sentiment":{
                "type":"negative",
                "score":"-0.654718"
            },
            "entities": [
            {
                "type": "Person",
                "text": "Bob"
            } ]
        },
        "action": {
            "text": "attacked",
            "verb": {
                "text": "attack",
                "tense": "past"
            }
        },
        "object":{
            "text":"beautiful Ana",
            "sentiment":{
                "type":"positive",
                "score":"-0.584455"
            },
            "sentimentFromSubject": {
                "type": "negative",
                "score": "-0.618875"
            },
            "entities": [ {
                "type": "Person",
                "text": "Ana"
            } ]
        },
    ]
}
```

Chapter 3

Implementation Details

In this section, we will discuss the implementation details of using the 3rd approach discussed in Section 2. The implementation has 3 phases

- Crawling
- Indexing
- Query and results

3.1 Crawling

For this project, we have developed crawlers for downloading data from different newspaper sites. Due to lack of available news data, we used the REST APIs for different newspapers sites and collected data. We designed crawler to download data with given date ranges. The crawler is designed for The Hindu, Indian Express, First Post, Times of India. Given a date range, all the crawlers gather data and output in an xml file for further processing.

For crawler design, we have used scrapy package in python[4]. Each article has title, content, url from which it is downloaded, and unique identifier. This xml is then given as an input to python program which invokes alchemy API for text content from each news articles. The format of xml is as follows:

3.2 Indexing

The next step after crawling is indexing. In this the json response from API is parsed and relevant information is extracted. This information is then stored in structured format in mongo-db[3]. This helps to index data based on article id. Mongo db is useful for storing and retrieval.

Each row contains following columns:

Listing 3.2 News data format

```
<xml>
    <articles>
        <article>
            <articleId>F1</articleId>
            <URL>the url for article</URL>
            <title>    title of article </title>
        <body> news article content </body>
        </article>
        <article>
        ...
        </article>
    </articles>
</xml>
```

Listing 3.3 News data format

```
articleId=F1
sourceEntity=Smriti Irani,
targetEntity=Jawaharlal Nehru University,
URL="..."
title="title of the article"
polarity=0.44
source="The Hindu"
```

3.3 Query Processing

We have developed a user interface as shown in figure 3.1 to enable searching and comparative analysis of opinions expressed in different news sources. The interface accepts source entity and target entity from user. The source entity is opinion holder and target entity is the one on which source entity has some opinion. The result contains the polarity of opinions from source to target.

The result is displayed for each article in which the opinion between these two entities is found. Result is also displayed for each news sources as shown in figure 3.2. This enables comparative analysis between different news sources over same issues.

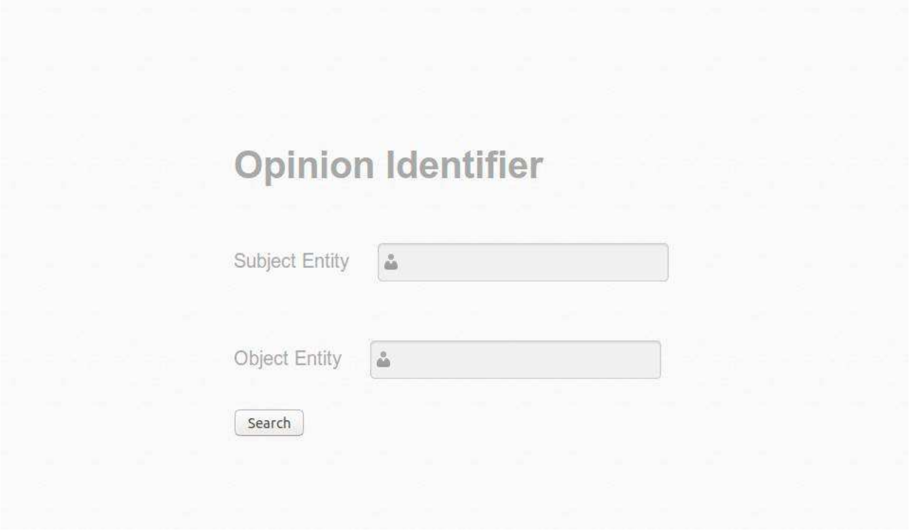
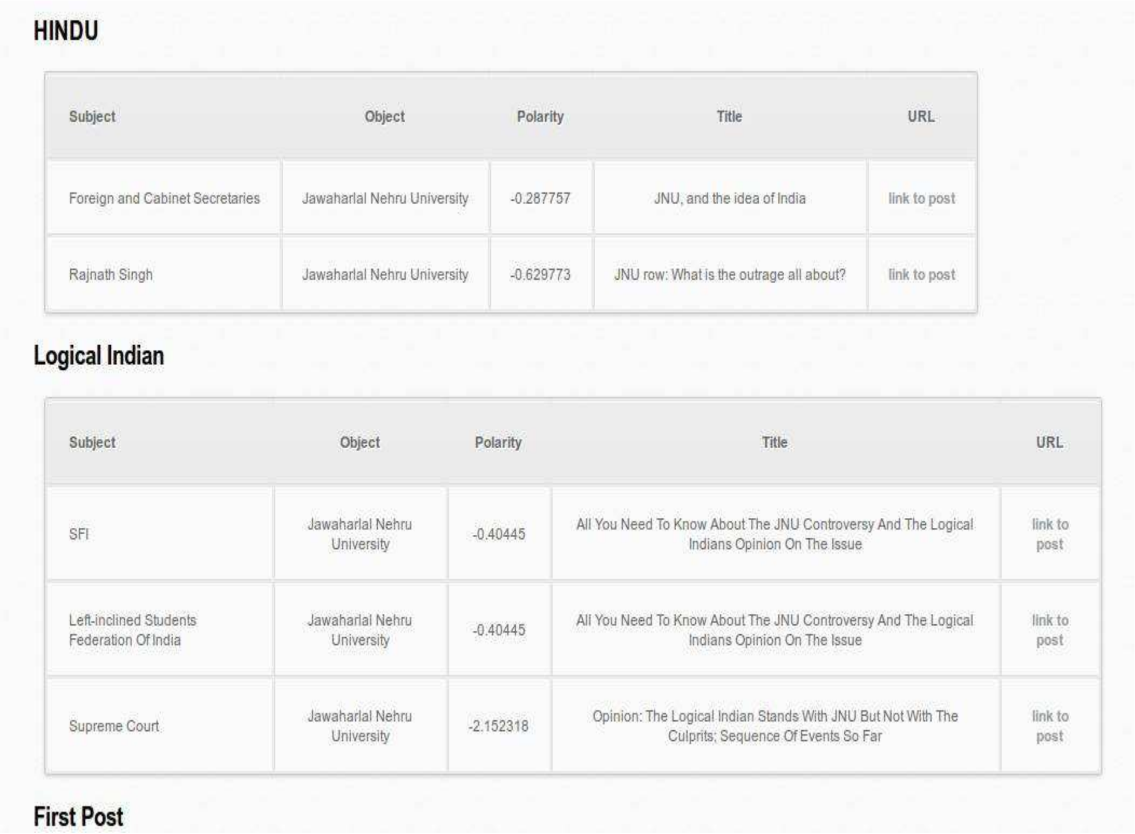


Figure 3.1: User interface



HINDU

Subject	Object	Polarity	Title	URL
Foreign and Cabinet Secretaries	Jawaharlal Nehru University	-0.287757	JNU, and the idea of India	link to post
Rajnath Singh	Jawaharlal Nehru University	-0.629773	JNU row: What is the outrage all about?	link to post

Logical Indian

Subject	Object	Polarity	Title	URL
SFI	Jawaharlal Nehru University	-0.40445	All You Need To Know About The JNU Controversy And The Logical Indians Opinion On The Issue	link to post
Left-inclined Students Federation Of India	Jawaharlal Nehru University	-0.40445	All You Need To Know About The JNU Controversy And The Logical Indians Opinion On The Issue	link to post
Supreme Court	Jawaharlal Nehru University	-2.152318	Opinion: The Logical Indian Stands With JNU But Not With The Culprits; Sequence Of Events So Far	link to post

First Post

Figure 3.2: Result

Chapter 4

Results and Analysis

In this chapter, we will see the entity-entity graph and entity relation graph obtained for the news dataset.

4.1 Entity-Entity Graph

We have plotted the entity-entity graph as shown in fig 4.1 which contains all the entities in an articles, somehow these entities are related to each other as they express opinions on same issues. This graph is undirected and indicates a cluster of entities that have opinion on same topic. If news articles are on different topics, we get multilple such entity clusters. We have plotted on smaller data so that it is easy to analyse.

From figure 4.1 we see that all the entities have opinion on JNU issues. The news are all on the same issues.

4.2 Entity-Relation Graph

This graph is directed graph. It is directed from source entity to target entity. The weight of the edge is the polarity between the two entities. The darker nodes have higher degree i.e. there are multiple edges starting from the node.

This graph is created on a smaller dataset to be able to analyze and visualize the relations in a better way.

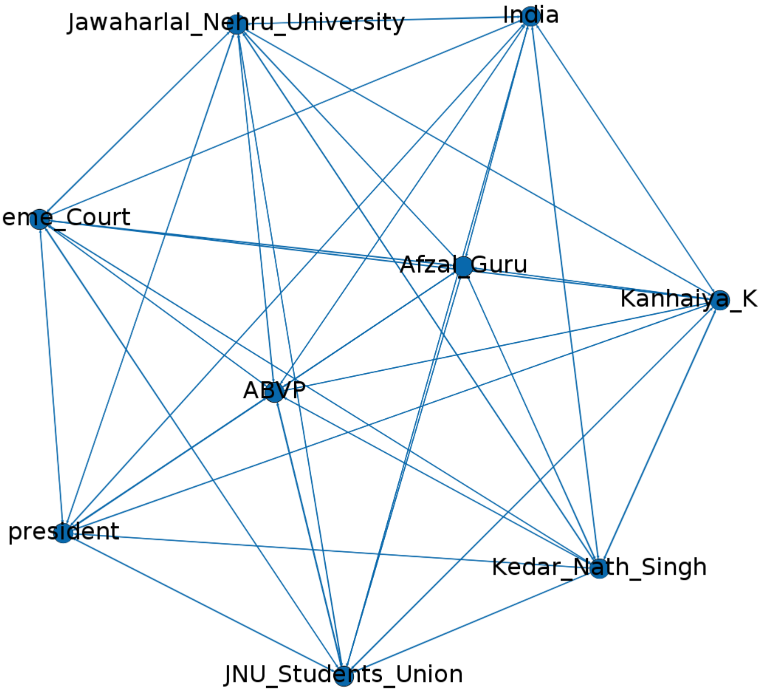


Figure 4.1: Entity-Entity graph

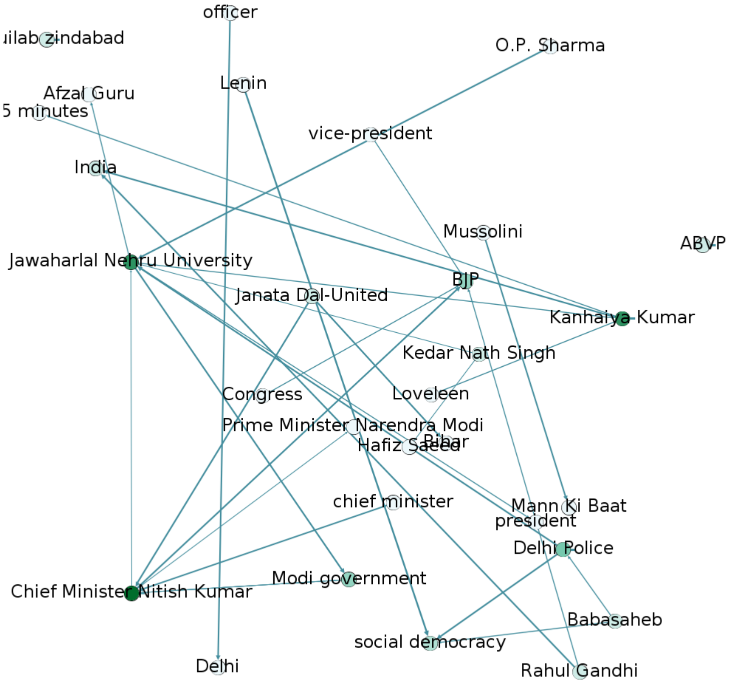


Figure 4.2: Entity-Relation graph

Chapter 5

Conclusion and Future Work

In this project, our aim of comparing the opinion sentiment from different news sources is achieved. We have crawled political and controversial news so as to provide comparative analysis but it can be extended to any category of news.

The UI is developed to get better analysis from different news sources. Currently, we have taken news mainly from The Hindu, Logical Indian and First Post, but can easily provide analysis for any number of news sources.

The entity-entity graph and entity relation graph is created to provide better visualization between different entities from news articles.

In future, we can extend this to extracting topics instead of entities and identify opinion holders expressing opinions on certain topics. This can be easily extended from the current model that we designed.

Chapter 6

Installations

1. Step 1: Crawling- AspectBasedSentimentalAnalysis/crawler/newscrawler/newscrawler contains crawlers coded for The Hindu, Indian Express and First Post. We have used scrapy package in python for crawling data. It downloads data based on the xpath mentioned. Similar crawler can be developed for different news sources. Basic changes that are needed for other sources:

   (a) Add the required domain name in allowed_domains

   (b) Add the root URL in start_urls

   (c) In parse method, provide the xpath of the content you need to download.

2. Step 2: Alchemy tool For this, data is available in the form of xml as discussed in previous sections.

3. Step 3: Node.js server setup Pre-requisite: ———— Install node js, mongo db, express framework of node js.

   (a) Run db.py to store the json extracted from alchemy in defined format.

   (b) Run command ‘npm install‘ to install packages required by node js

   (c) Run command ‘npm start‘ to start the server listening in port 3000

   (d) Open browser and hit the url ‘localhost:3000‘ to reach main page

   (e) Two textfields are present. One is for subject entity and other is to specify object entity

   (f) Results contains following fields: Subject Entity Object Entity Polarity Title (of the article expressing this opinion) Link to article

   (g) Polarity is a real number whose value represents the intensity of positiveness or negativeness of the opinion. 0 represents neutral opinion.

   (h) Both fields are optional. Based on entered value, system behaves in following way

      - Subject only : Opinion of subject on any topic or any other entity will be extracted
      - Object only: Opinion from anybody/ any entity to that object will be extracted.

- Subject + Object: Opinion from specified subject entity to Object entity/topic will be extracted.

  None: It shows all data in db for the crawled dataset

(i) All results are displayed in three seperate sections categorized based on news source. ’The Hindu’, ’FirstPost’, ’Logical Indian’

Bibliography

[1] http://nlp.stanford.edu/projects/coref.shtml.

[2] http://nlp.stanford.edu/software/stanford-dependencies.shtml.

[3] https://www.mongodb.org/.

[4] Scrapy tutorial. http://doc.scrapy.org/en/latest/intro/tutorial.html.

[5] Textblob: Simplified text processing. https://textblob.readthedocs.io/en/dev/.

[6] Sentiment Analysis API. http://www.alchemyapi.com/products/alchemylanguage/sentiment-analysis.

[7] AlchemyAPI: An IBM company. Alchemy api. http://www.alchemyapi.com/.

[8] Soo-Min Kim and Eduard Hovy. Extracting opinions, opinion holders, and topics expressed in online news media text, July,2006. Proceedings of the Workshop on Sentiment and Subjectivity in Text.

[9] Steven Skiena Namrate Godbole, Manjunath Srinivasaiah. Large-scale sentiment analysis for news and blogs, 2007. ICWSMâĂŹ2007 Boulder , Colorado, USA.

[10] Toyoaki NISHIDA Tomohiro FUKUHARA, Hiroshi NAKAGAWA. Understanding sentiment of people from news articles: Temporal sentiment analysis of social events, 2007. CWSMâĂŹ2007 Boulder, Colorado, USA.