Political Actor Recommendation System

Harsh Bhavsar, Harshal Mehta, Khushbu Patil, Mittal Patel, Shruti Shetye

Department of Computer Science

The University of Texas at Dallas

(hhb140330, hcm140230, kxp153130, mdp140430, sns150630)@utdallas.edu

ABSTRACT

The traditional dictionary based pattern lookup approach to identify political actors. The current implemented traditional approach focuses on who-did-what-to-whom format. Several drawbacks can be encountered with the current implementation since it is manually dependent for it's updating. This dependency brings in other drawbacks like infrequent updating and high maintenance cost.

The proposed framework of Political Actor Recommendation is inspired by and based on the paper published – “Discover New Actors In Politics: A Framework To Recommend Political Actors With Role In Real-Time” by Mohiuddin Solaimani, Sayeed Salam, Latifur Khan, Patrick Brandt, Vito D’Orazio.

# **1. INTRODUCTION**

Encoding of political data can be done by humans or machine learning. Dictionaries of actors and actions are maintained. For useful processing and analysis there is a strong need of processing political data to extract useful information such as politicians and their roles in various organization. Identification and addition of new political actors and their roles into the database, when done by humans it has high precision because of the human brain interpreting the data. However that process is very slow and inefficient. Recommending an actor with the role based on preexisting data creates an issue when the information about that actor is not updated in the dictionaries yet. There are machine learning based semantic labeling techniques to identify verbs, nouns, etc. Using these techniques we can identify names of political actors and their existing roles too.

# **2. TECHNOLOGY STACK**

Apache Kafka, Spark Streaming library, NoSQL (Redis), Stanford NLP, Stanford NER, CAMEO action directory, various APIs for obtaining live news.

# **3. FRAMEWORK**

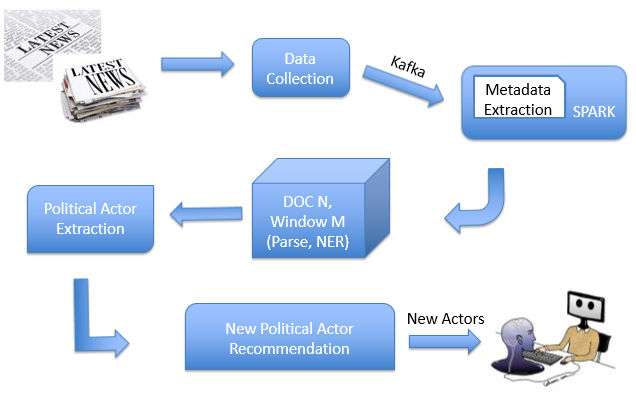
  
**Figure 1: Overview of Framework for Actor Recommendation**

Figure 1 shows our proposed framework for political actor recommendation. It collects political news stories periodically through News API which provides data and news from several news sources. Later, it uses CoreNLP to extract metadata from the stories like parse trees, tokens, NER, semantic events, etc. and stores them in our database which is built using NoSQL. We use Apache Spark streaming with Kafka to collect all the scrapped data, periodically. Here, we used the time interval to be 2 hours. We use CoreNLP inside a Spark worker node to scale up the process. After that, it classifies political events using own data structure (which is built considering CAMEO as a base) from the metadata. It fetches possible new actors from source and target of the events and NER. Finally, human experts validate the recommended actors with roles and update the dictionary. We describe our framework in details in following subsections.

**3.1 Data Collection**

We are using NY Times API for collection of political news articles and information. We use these chunks of data to extract useful information and main content of the news. We request data from the API periodically and it returns JSON metadata for the headlines currently published on a range of news sources and blogs. This data is handles using apache kafka and handed over to Spark.

**3.2 Political Actor Extraction from JSON Metadata**

Inside our data processing unit, CoreNLP parses the data and extracts metadata like Parts-Of-Speech (POS) tagging, Parse Tree, Named Entity Recognition (NER) etc. and stores them into our database. We use NER for our framework.  
**NER (Named-entity Recognition):** Locates and classifies label sequences of words in a text which are the names of things, such as person and company names, or gene and protein names. In our framework, we use persons and organizations to detect new actors. We use NER given by Stanford CoreNLP because it has higher accuracy than other similar tools like Alchemy, OpenCalais, OpenNLP, etc.

**3.3 New Political Actor Recommendation**

First we aggregate the actor names with multiple aliases. To do this we have implemented Jaccard Distance based K-means clustering. Once grouped, the actors are compared in the Redis database dictionary if they have a very high rank (based on frequency of occurrence). If the name is not already in our database then we add it. Then we recommend the actors based on ranks.

**3.4 Human Feedback**

During the run of the program, using human intervention, we can discard some cases and give valuable feedback to the overall system.

# **4. TECHNICAL VIEW POINT**

# 

**Figure 2. Technical View**

Figure 2 describes technical details of the framework. First we scrape the news article data using NY Times API. Then it is channelized to kafka producer. NY Times API gives JSON metadata about political articles. Kakfa consumer takes that data and feeds it to spark streaming. Then we periodically stream the data and use NER to extract the political actor names. There can be a person with multiple different aliases. To address this issue, we group the actor names together using Jaccard Similarity. The names with highest similarities will most probably be same person. Then we catalog each name and find its frequency of appearance in the articles and rank them. We also have implemented our own CAMEO like dictionary using NoSQL database. In the dictionary we have implemented n-to-n mapping. Whenever a new actor with high rank comes, we add it to the dictionary if the actor is not already present.

Scenario: For example let's consider a certain batch of news article that has Donald Trump in it for a certain role. Now the system can't possibly recommend it because there is no pre-existing information about him and his new role. So if our system encounters his name with a high rank (frequency of appearing in documents) then it'll check with our NoSQL-CAMEO-like dictionary and add it to the dictionary if needed. This way in subsequent runs the dictionary is now updated and can be used for recommendations of those new actors as well.

## **5. FUTURE SCOPE**

Our approach can be extended further to include a Graphical UI and also a built-in framework for role-discovery and appropriate role prediction for actors as well. Similarity computation amongst the names can be improved to eliminate the possibility of aggregating actors with similar names (E.g. Bill Clinton, Bill Nye).

# **6. REFERENCES**

* *Discover New Actors In Politics: A Framework To Recommend Political Actors With Role In Real-time*   
  *Paper by : Mohiuddin Solaimani , Sayeed Salam , Latifur Khan , Patrick T. Brandt and Vito D’Orazio*
* *Stanford NER -* [*https://nlp.stanford.edu/software/CRF-NER.shtml*](https://nlp.stanford.edu/software/CRF-NER.shtml)
* *Spark Streaming - spark.apache.org/docs/latest/streaming-programming-guide.html*
* *News API -* [*https://newsapi.org/*](https://newsapi.org/)
* *NY Times API -* [*https://developer.nytimes.com/*](https://developer.nytimes.com/)
* *Redis – https://redis.io/*
* *Kafka -* [*https://kafka.apache.org/*](https://kafka.apache.org/)
* *CAMEO - http://eventdata.parusanalytics.com/data.dir/cameo.html*

.