Connecting Dots between News Articles

CS365 Project Proposal

Under Guidance of Prof. Amitabha Mukerjee

Group 15 : Ankit Modi (10104), Chirag Gupta (10212)

Introduction

Given a set of articles, a source article and a target article, we have to find a coherent chain of articles linking them together. That is, somehow the chain formed should appear to be in a logical and consistent order comprising a story.

Motivation

This project aims at tackling the problem of **information overload**. With huge amount of data being published daily, users may miss the bigger picture of a story. We would focus on the news domain as it covers nearly all spheres of an individual's interests like politics, sports etc.

If user is interested in knowing the particulars about a specific story, then scrolling the whole newspaper, and that too of several days, becomes a painstaking task. This method would help the user to navigate easily between topics.

Related Work

The problem has been solved by shortest path algorithm in previous researches but in that there isn't any focus on the coherency of the output.

One approach for avoiding this is using influence of word on the articles [1]. This takes into account both missing words between articles and the variable importance of some words than others.

A different approach for calculating coherence is mentioned in [2]. Here the concept of m-coherence is introduced ie. an m-coherent chain is one in which each subset of length m is coherent.

Other related work in this field has been done on identifying and tracking news events, narrative generation which might be of use to us.

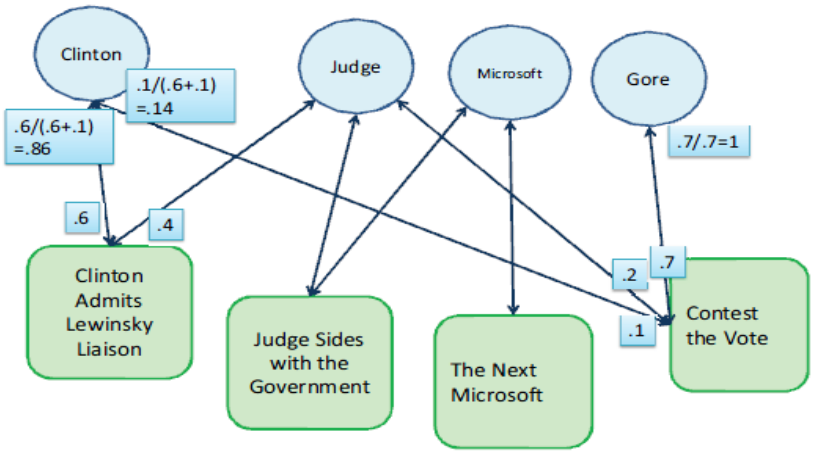
Resources

- Dataset: We are still looking for a relevant dataset.
- Code: We will be developing our own code to implement this research. Codes of previous work haven't been released yet.
- TF-IDF (Term frequency – Inverse document frequency) library in python: https://code.google.com/p/tfidf/

Our Implementation

- We will follow the implementation of paper [1] mainly. Our implementation language will be Python.
- Our dataset would be a chronologically ordered set of articles. (D_1, D_2…D_n).

- We construct a bipartite graph between articles and words. All our further calculations will be done on this graph. Figure credits: Paper [1].



A bipartite graph used to calculate influence.

- *Edge weights* (TF-IDF will be used for this) will be added to this graph representing strength of relation between articles and words. In this the importance of a word is taken into account. Also some words missing in one or two of these articles can still be an important link connecting the two articles.
- These will be helpful in evaluating *Influence* of a word in connecting two articles. They are a better measure than simple word count.
- Then we shall use a Linear Programming model having four modules namely *Chain restriction*, *Smoothness, Minimax Objective* and *Activation Restrictions* to attain the objective. These are referenced from the paper [1].
- Coherence, Relevance and Redundancy will be the evaluating parameters for our output chain of articles.

References

[1] Dafna Shahaf and Prof. Carlos Guestrin : Connecting the dots between news articles. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) 2010.*

[2] Dafna Shahaf , Prof. Carlos Guestrin and Eric Horvitz : Trains of thought-Generating information maps. *International World Wide Web Conference (WWW), 2012*.

[3] Wikipedia:

http://en.wikipedia.org/wiki/Bipartite_graph

http://en.wikipedia.org/wiki/Tf%E2%80%93idf