CONNECTING THE DOTS BETWEEN NEWS ARTICLES

Ankit Modi (10104) and Chirag Gupta (10212)

Guide: Prof.Amitabha Mukerjee

Abstract

One of the major technical challenges being faced by us these days is of *Information Overload*. There is an enormous amount of data being thrown at us almost every day. This has impacted our ability to see the bigger picture in a negative way. Also, we are facing lots of difficulties in navigating between different topics and unravelling the hidden connections between them. This project aims at tackling this problem.

Acknowledgement

We would like to acknowledge the contributions of some of the individuals who went out of their way to help us complete this project.
First of all, we would like to thanks *Dafna Shahaf* and *Prof. Carlos Guestrin* for writing the mentioned paper which academically stimulated us to take up this project. Special thanks to *Dafna Shahaf* for writing to us and explaining our doubts with utmost patience. We really appreciate the pain she took to write us back.
We are very grateful to *Prof. Amitabha Mukerjee* for guiding us throughout this project. Apart from his inputs and suggestions which helped us improve our work drastically, he has also played a keen role in keeping us motivated throughout our project.
Lastly, we are also thankful to all the TAs and our course mates who gave us their invaluable insights regarding our work.

Previous Work

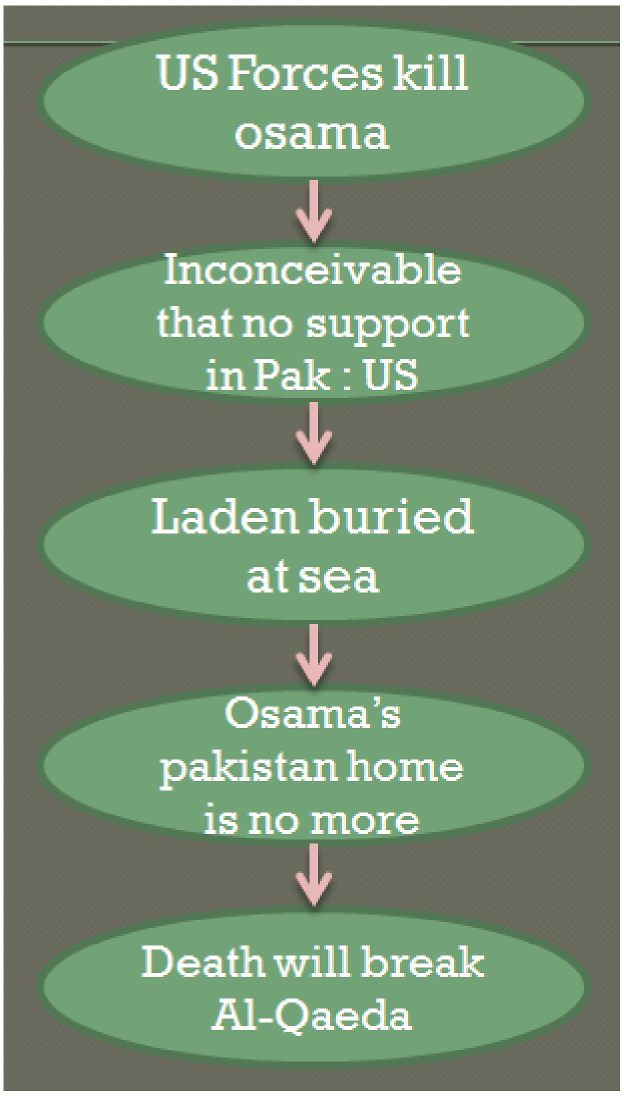
The major inspiration of this project comes from the paper published in 2010, *"Connecing the dots between news articles" by Dafna Shahaf and Prof. Carlos Guestrin.* We have focussed on the news domain as news browsing is one of the primary uses of the internet and it covers almost every aspect of an individual's life namely sports, politics, entertainment etc. Moreover, searching for relevant news has become a cumbersome task. Our project can always be extended to other domains like research papers etc.

Introduction

Given a set of documents, a source and a target document, we aim at generating a coherent chain linking the source and target documents into a meaningful story. For example, it can discover the chain between Jessica lal murder case and Nitish katara murder case.

As we have selected the news domains, given two news articles, our aim is to generate a coherent chain linking them together.

For example, given a source *'US forces kill Osama'* and a target *'Osama's death will break Al-Qaeda'*, the chain generated is


US Forces kill osama
Inconceivable that no support in Pak : US
Laden buried at sea
Osama's pakistan home is no more
Death will break Al-Qaeda

Generation of chain is followed by its evaluation.

We evaluate the characteristics of a good chain by its *coherence.* We have used two methods to find the coherence of a chain. One of them (called *coherence1*) has been taken straight from the above mentioned paper of *Dafna Shahaf* where as for the second part (called *coherence2*), we have improvised upon the previous method to see if it gives any better any results.

Finally, we compare different combinations of weight-assignment methods and shortest path algorithms to see which combination gives us a better coherence.

Our Work

Phase I : Chain Computation:

Approach 1 :

We model the documents as normalized histograms with words on the x-axis and word count in a document as the y-axis and height of the histograms.

Then for a similarity measure between documents we use the Bhattacharya's distance that is defined as:

$$DB = -\ln(BC(p,q))$$

where $BC(p,q) = {}_{x\in X}\Sigma\,(p(x).q(x))^{1/2}$ *is the Bhattacharyya coefficient*

We have used x as the common words between the two articles.

p(x) is the number of occurrences of x in document p divided by total number of words in document p.

q(x) is used similarly for document q.

Bhattacharya's distance thus calculated is closer to 0 for more similar articles and farther than 0 for dissimilar articles due to the –ve logarithm taken of the Bhattacharya's coefficient.

Using this Bhattacharya's distance as the edge weights between two documents, we model the entire dataset as a graph, and calculate the adjacency matrix for it.

On this adjacency matrix, when we get the source and target articles we compute the shortest path using one of the shortest path algorithms out of Dijkstra's algorithm and A* algorithm.

Approach 2

In this approach, first of all, instead of Bhattacharya's distance we use TF-IDF weights to calculate the effect of a word on an article. TF-IDF or Term Frequency –Inverse Document Frequency is defined as:

tfidf (t,d,D) = tf (t,d) X idf (t,D)

Where

tf(t,d)= No. of occurrences of term t within document d divided by total no. of words in document d

and **idf(t,D)** = logarithm of no. of documents divided by no. of documents with term t

So, for each document we calculate these weights for each word, and model the document as vectors with these word weights as their components.

Then we compute the cosine similarity between these document vectors.

The similarity is just the dot product of the two vectors divided by their magnitudes.

The more similar two articles are the greater the value of cosine similarity is and since it is a dot product the value of this similarity varies from 0 to 1, ie. It gives us the percentage similarity between two articles.

We use the reciprocal of this similarity as edge weights between any two documents of the dataset.

Again modelling these distances as an adjacency matrix, we compute the shortest path between the source and the target using both Dijkstra and A* algorithms.

Phase II : Chain Evaluation

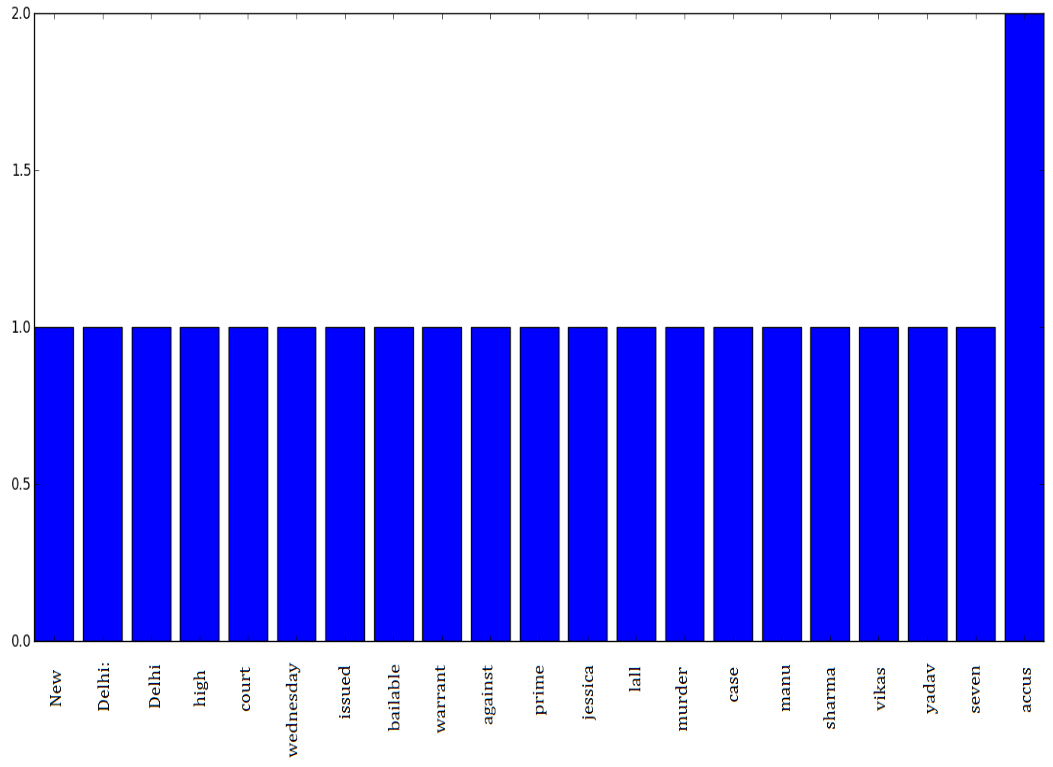
Now that we have a chain, we evaluate our results using coherence of the output chain as a parameter. We also use the fact that a chain is only as strong as its weakest link. For this we use two evaluation measures:

- **Coherence1**(d1, …,dn) = $_{i=1\ldots n-1}\min \Sigma_w 1(w € d_i \cap d_{i+1})$
  This is the minimal transition score .
  Ref : Dafna Shahaf and Prof. Carlos Guestrin : Connecting the dots between news articles. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) 2010.*

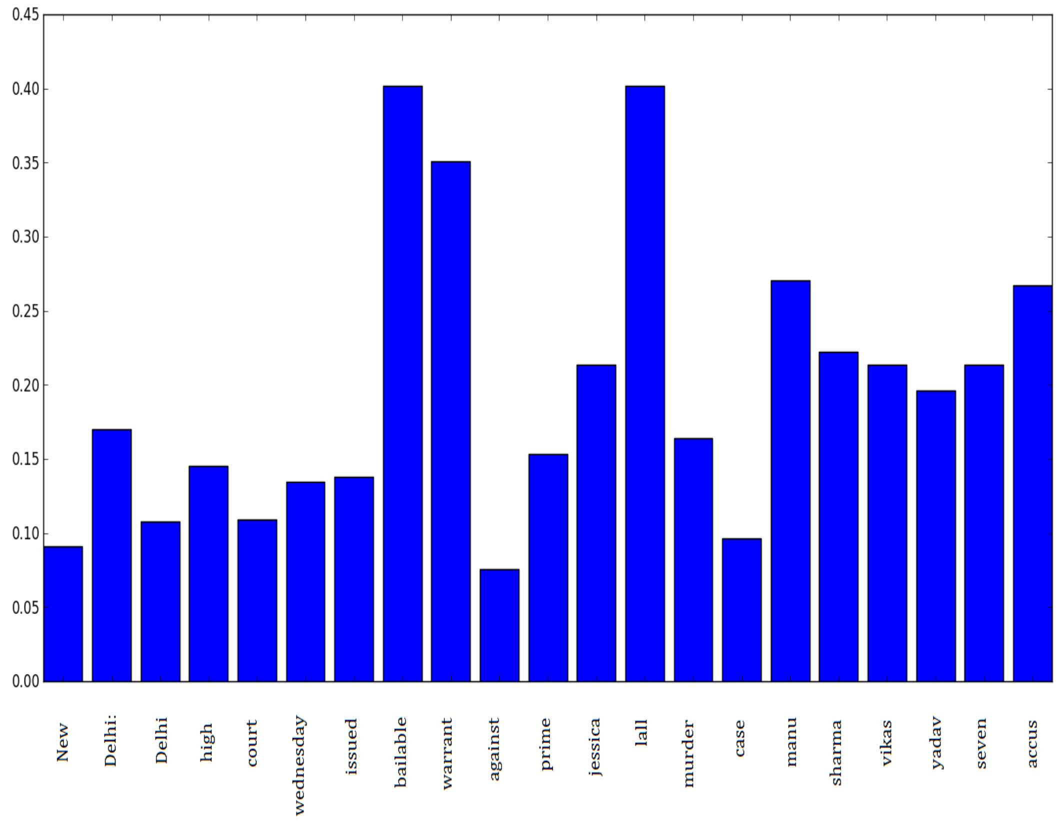
- **Coherence2**(d1, …,dn) = $_{i=1\ldots n-1}\min$ { cosine-similarity (d_i , d_{i+1}) }
  This measure guarantees a value between 0 and 1, and hence, can be used as the percentage similarity between documents.

Conclusions

- ***TF-IDF is better than unigram***



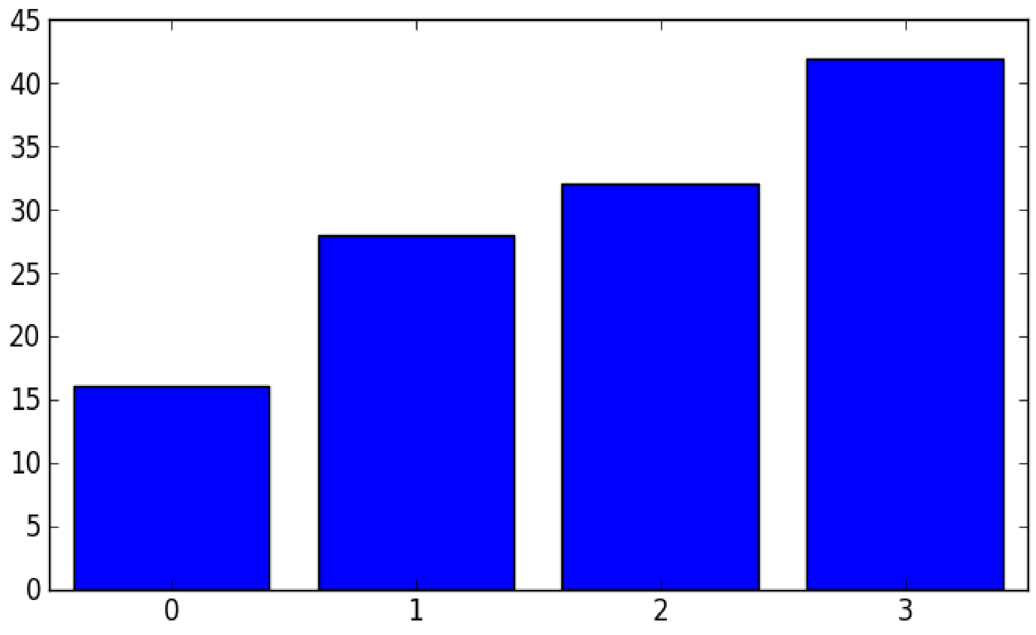
Unigram count in an article, the corresponding words are shown on the x-axis. All words with equal count are weighed equal.



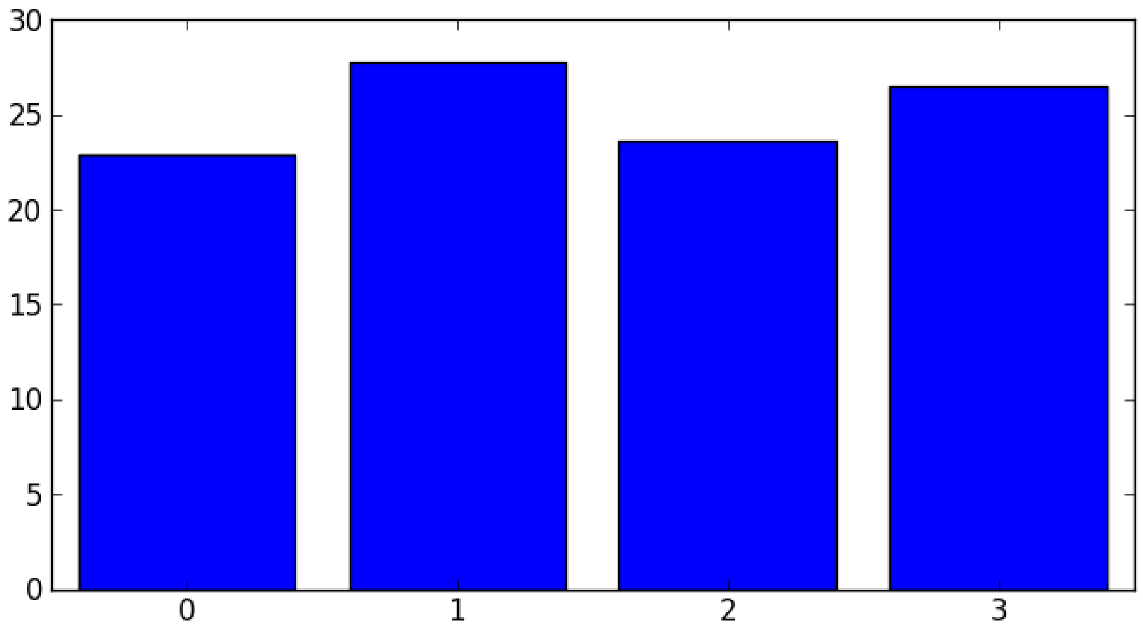
TF-IDF weights for words on the same article. This increases the weight of words like "Jessica" and "Lall" and decreases the weightage of words like "against".

- ***Coherence2* is a better evaluating parameter than *coherence1***

  We took a chain quite coherent to us and evaluated both Coherence 1 and coherence 2 on it.



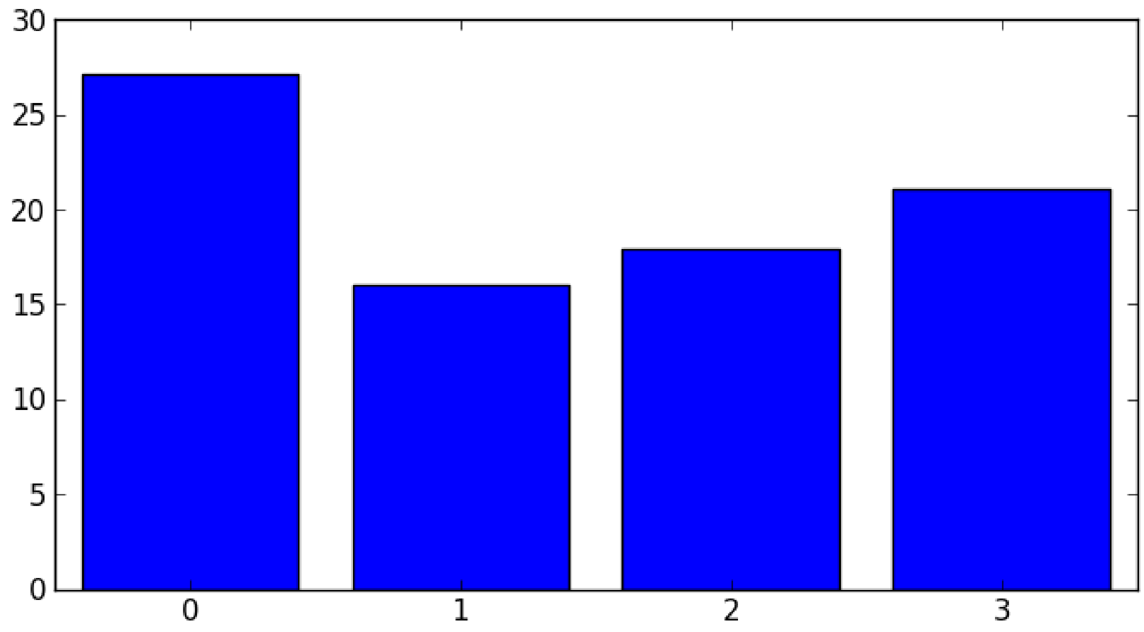
Coherence 1’s graph (above) is clearly not rating the coherent chain well. There is a lot of variation between adjacent chain values from 16 to 42.



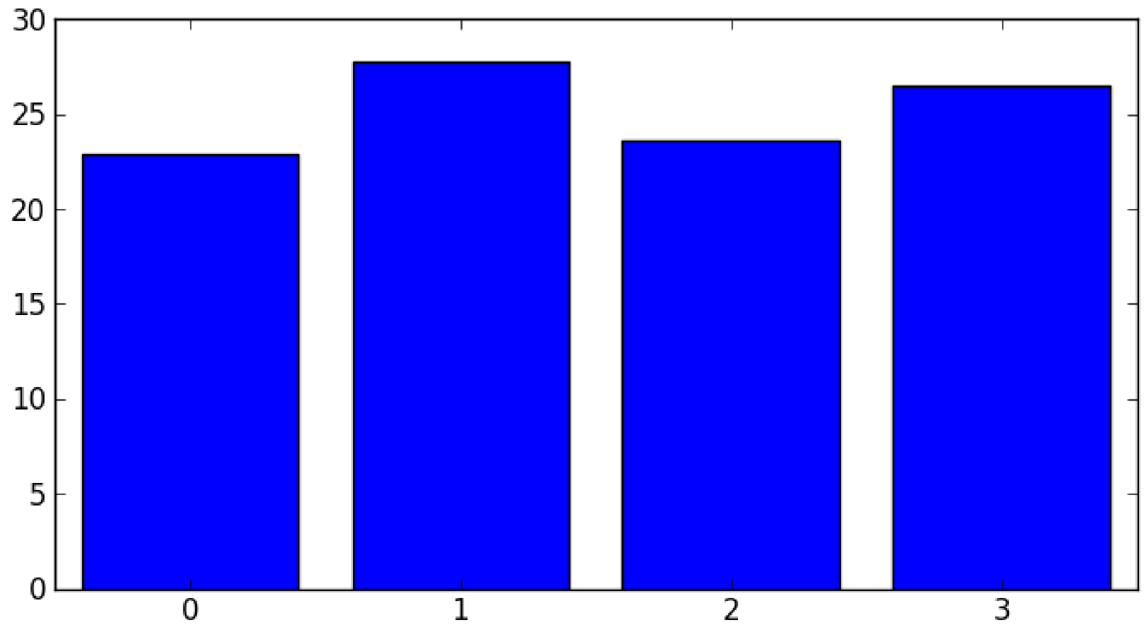
On the other hand coherence 2, rates all values between 22 to 27 and is thus evaluating our chain as coherent.

- ***Cosine similarity using TF-IDF gives more coherent chain than Bhattacharya’s distance.***

  For this, we used coherence 2 as a parameter for evaluating the output chain.



This figure corresponds to the chain formed by Bhattacharya distance evaluated by coherence 2. Clearly the value of the weakest link’s strength is 15.



Whereas by using cosine similarity measures, the value of the coherence 2 comes out to be 22.

Tabulated Results on the same source and target article for different combination of methods.

Weight	Algorithm	Chain	Coherence1	Coherence2
Bhatt. Dist	**Dijkstra**	**123-117-105-113-111**	**11 (105-113)**	**16.05% (105-113)**
Cosine-Similarity	**Dijkstra**	**123-103-102-110-111**	**16 (110-111)**	**22.88% (110-111)**
Bhatt. Dist	**A***	**123-118-103-117-111**	**18 (117-111)**	**19.24% (118-103)**
Cosine-similarity	**A***	**123-103-117-110-111**	**16 (110-111)**	**20.45% (117-110)**

The two shortest path algorithms don't show much of a difference. Also another observation from the above table is that even coherence 1 rates the cosine similarity method more than the Bhattacharya's distance for the Dijkstra algorithm case, and since the two coherence 2-values in A* algorithm are nearly the same, coherence 1 gives a different order.

Overall, we try to claim that Cosine similarity and Coherence2 are the better chain calculation and evaluation method respectively than Bhattacharya's distance and Coherence1.

Future Work:

- Notion of *m-coherence* can be used instead of *coherence* for better evaluation of results.
  Ref: Dafna Shahaf , Prof. Carlos Guestrin and Eric Horvitz : Trains of thought-

Generating information maps. *International World Wide Web Conference (WWW), 2012.*

- Using different corpuses (other than News articles) may help in important scientific discoveries.
- With millions of articles being produced daily worldwide, this process needs to be implemented onto a large scale.

Bibliography

[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] Michael D. Lee, Brandon Pincombe and Matthew Welsh : An Empirical Evaluation of Models of Text Document Similarity. In Proceedings of the 27th Annual Conference of the Cognitive Science Society (2005).

[4] Deept Kumar, Naren Ramakrishnan, Richard F. Helm, and Malcolm Potts : Algorithms for Storytelling. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 20, NO. 6, JUNE 2008

[5] M. Shahriar Hossain, Joseph Gresock, Yvette Edmonds, Richard Helm, Malcolm Potts and Naren Ramakrishnan. Connecting the Dots between PubMed Abstracts. 2012