Week1:

Strengths :

1. **Recognition of crowd sourced knowledge repositories as autopoietic systems.**
2. **Identifying interlinks as potential factoids.**
3. **Studying the factoids in a systematic way using timestamps and RFFR lists.**
4. **First of its kind in studying the phenomenon of triggering on real world data.**
5. **Highlighted the triggering phenomenon using factoids showing the similar factoids being entered in successive revisions.**

Weaknesses :

1. **Not considering the factoids that got extinct as some of them might have given rise to new factoids that are present in the final article. Their contribution is not accounted for.**
2. **NGD value can be small even if the phrases are very dissimilar when the information about them is limited on internet. h(a) and h(b) would be small if number of google results are small, giving an overall small value.**
3. **Threshold value considered is very dependent on the article chosen.**
4. **Small NGD value may not mean triggering in all cases.**

Further Scope:

1. **Considering contribution made by the extinct factoids.**
2. **Expanding the current representation of factoids (internal links) to encompass multiple crowdsourcing portals.**
3. **Different articles are related to each other and thus they are also responsible for knowledge building of each other. We should consider that also.**

Week 2:

Increase number of Calls for NGD

- Google api restricted to limited number of uses - 100 queries per day.
    1. Additional Charges - 5$ per 1000 queries, upto 10k queries per day.
    2. Custom Search Site Restricted JSON API that has no daily limit. But it limits the search results only to 10 or lower sites.
- Hoax Calls not Possible. Also Against the Google TOC.
- Can use curl and wget command to download pages, and then use python to scrape HTML documents. But Google temporarily blocks IP once automated calls are made in succession.

Various other measures to analyse the semantic similarity of two different phrases

- **Latent Semantic Analysis (LSA):**
  http://lsa.colorado.edu/whatis.html
  The key idea is that for a given word or phrase, all the possible contexts in which it can be used provide certain constraints, which help us in determining the similarity of different sets of words.
- **Explicit Semantic Analysis:**
  http://www.cs.technion.ac.il/~gabr/papers/ijcai-2007-sim.pdf
  Here the concepts and knowledge of wikipedia are used as the dimensions of a vector space, based on the weights of these dimensions we describe certain words or paragraphs as vectors. The similarity can then be measured using conventional metrics such as cosine.
  Text input is represented as the vector of concepts.
- **Word2vec:**
  Shallow models with two layer neural networks, trained to represent linguistic contexts into vectors.

Week3:

Word2vec:

- We can use word2vec model for measuring the similarity of two phrases.
- Any phrase can be seen as a collection of words and for each word we can have some top similar words and doing so for each word we can form a bag of words corresponding to one phrase, similarly for the second phrase we can get a bag of words.
- Now we can get their similarity measure by comparing these two bags.
- However this method is very time consuming and the comparing of two bags does not always guarantee that their semantic meaning is also compared.
- Moreover the model has a limited vocabulary of around 3M words, and the results are highly dependent on the training dataset.

Latent Semantic Analysis:

- Take documents.
- Extract words leaving out redundant words.
- Create a matrix with W*D dimension with entries as count of words.
- Apply SVD decomposition on this matrix and pick some dimensions thus leaving out noise.
- Corresponding to each document we got a vector.
- Can compare different documents using cosine of angles and their correlation.

Examples :
Nickelodeon and Genghis Khan: 0.05320959919727142
Grapes and Cars: 0.09600525428306506
Brad Pitt and Angelina Jolie: 0.476905537641647
Barack Obama and Michelle Obama: 0.5983324765267662

GOOGLE API

The learned models can be used to get a similarity measure for different measures, but it won't be that much effective as compared to the NGD. The high dependency of the models on the training data sets limits the semantic context to the data set only. So we have planned to use the NGD values as well.
For the calculation of NGD we will be using the Google Custom Search API, which allows 100 free hits per project, we can create multiple projects per account and create one Custom Search Engine.
We can create as much projects as allowed and and get their API keys which will be used during the REST API GET requests.
However the results given by the google Custom Search API are different as compared to what is actually displayed during search. (This is for all the phrases, thus we can rely on the API)

Google Scraping

- Scraping of Search engine allows us to search automatically, without letting the provider to know the calls are automated.
- Our problem can be implemented by scraping Google Search engine and then scraping the count values obtained.
- Against Google TOC.
- Can be used to search for innumerable number of pairs without any query limit.
- The count obtained is exact as found in manual searching unlike those obtained from Google API hit.

Extensions:

- As per now our goal is to get the factoid network with all the extinct as well as present factoids, with the increased NGD calculation we can get all the edges and their weights.
- Along with the timeline of the factoids, we can visualize through the graph how the network evolves with time.
- We can then study the formation of clusters and communities if any, in the factoid network.
- With time we can identify the how factoids behave, how they saturate and how some get extinct with time.