TOPIC SUMMARIZATION

TEAM : SUMMARY JACK

Junaid Shaikh
Mitali Vijay Bhiwande
Tariq Siddiqui
Tejasvi Balaram Sankhe

CSE 535 : Information Retrieval
Fall 2016
University at Buffalo
The State University of New York, Buffalo

Features:

A Tweet Topic Summarizer.

Based on twitter data corpus

Summarizes tweets obtained over ten thousand tweets.

1. Overview:

Our system works with the twitter data indexed on solr to provide sub-topic modelling for the same and generates summaries based on user inputs and queries for the retrieved tweets.

2. Data Accumulation:

Input tweets for this project were collected from Twitter using twitter API.
Tweet Collection:
Tweet Collection was performed twice. Once on the hashtag, #Demonetization with over 1000 tweets and finally on the hashtag, #Christmas with over 12000 tweets.
The java code written for tweet data collection is robust to API limitation and wouldn't easily waste the calls to API without returning any results and was completely automated.
The task of stopword removal was performed in this java code itself and a final json file was generated which was then fed as input to LDA for topic modelling. Removal of stopwords was performed so that the LDA model wouldn't model the redundant, repeated data.

3. Topic Modelling using LDA:

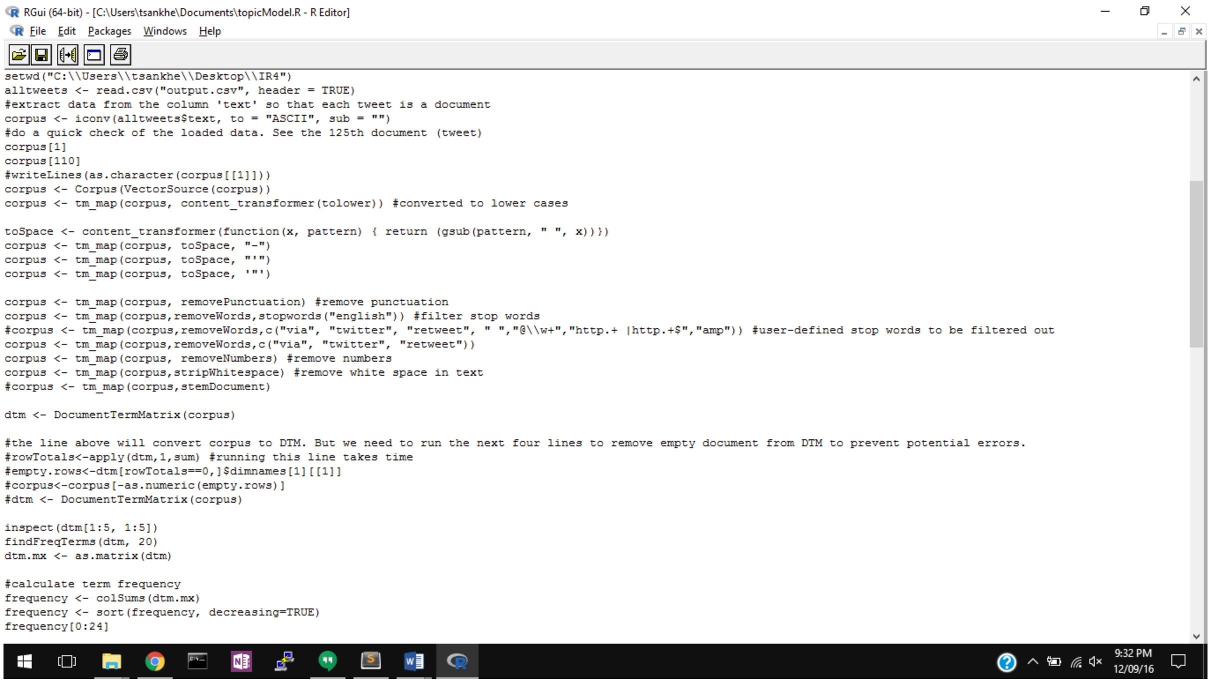
Our goal was to group the twitter corpus we collected into different topic groups and later generate subtopics for each topic, based on what the users were tweeting.

For this we decided to go with LDA topic model i.e. Latent Dirichlet allocation. LDA is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. Since tweets are an unstructured dataset using Latent Dirichlet Allocation which is based on machine learning would prove beneficial.

LDA is an unsupervised machine learning technique which identifies latent topic information in large document collections. It uses a "Continuous bag of words" approach, which treats each document as a vector of word counts. Each document is represented as a probability distribution over some topics, with each topic as a probability distribution over a number of words. LDA defines the following generative process for each document in the collection:
1. For each document, pick a topic from its distribution over topics.
2. Sample a word from the distribution over the words associated with the chosen topic.
3. The process is repeated for all the words in the document

Each document in the collection is associated with a multinomial distribution over T topics, which is denoted as θ and each topic is associated with a multinomial distribution over words, denoted as ϕ. Both θ and ϕ have Dirichlet prior with hyper parameters α and β respectively. K gives the number of topics.
In our case, LDA gives 5 topics i.e. K=5. After studying research papers this was found to be the optimum number of topics produced by LDA, when working with tweets corpus. We used R tool for LDA implementation.

Following is the code snippet:

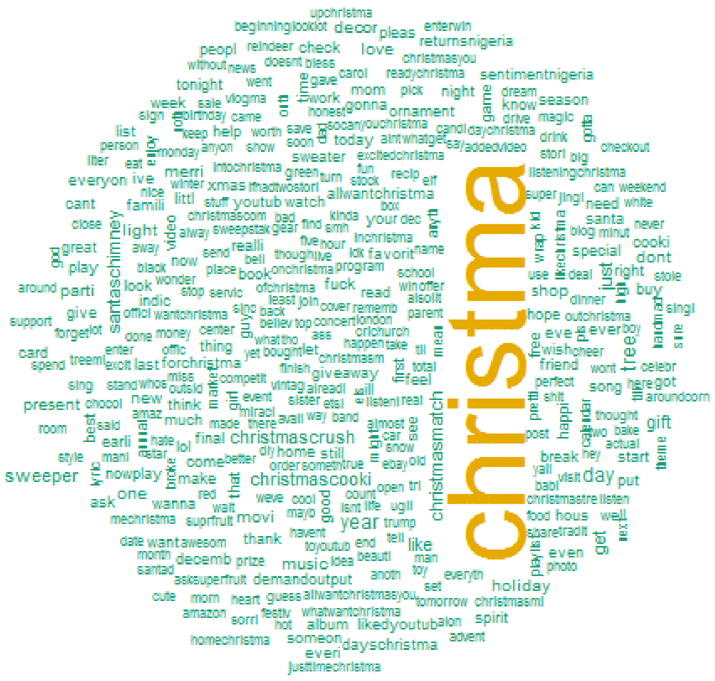


RGui (64-bit) - [C:\Users\tsankhe\Documents\topicModel.R - R Editor]
File Edit Packages Windows Help
9:32 PM
12/09/16

```
setwd("C:\\Users\\tsankhe\\Desktop\\IR4")
alltweets <- read.csv("output.csv", header = TRUE)
#extract data from the column 'text' so that each tweet is a document
corpus <- iconv(alltweets$text, to = "ASCII", sub = "")
#do a quick check of the loaded data. See the 125th document (tweet)
corpus[1]
corpus[110]
#writeLines(as.character(corpus[[1]]))
corpus <- Corpus(VectorSource(corpus))
corpus <- tm_map(corpus, content_transformer(tolower)) #converted to lower cases

toSpace <- content_transformer(function(x, pattern) { return (gsub(pattern, " ", x))})
corpus <- tm_map(corpus, toSpace, "-")
corpus <- tm_map(corpus, toSpace, "'")
corpus <- tm_map(corpus, toSpace, '"')

corpus <- tm_map(corpus, removePunctuation) #remove punctuation
corpus <- tm_map(corpus,removeWords,stopwords("english")) #filter stop words
#corpus <- tm_map(corpus,removeWords,c("via", "twitter", "retweet", " ","@\\w+","http.+ |http.+$","amp")) #user-defined stop words to be filtered out
corpus <- tm_map(corpus,removeWords,c("via", "twitter", "retweet"))
corpus <- tm_map(corpus, removeNumbers) #remove numbers
corpus <- tm_map(corpus,stripWhitespace) #remove white space in text
#corpus <- tm_map(corpus,stemDocument)

dtm <- DocumentTermMatrix(corpus)

#the line above will convert corpus to DTM. But we need to run the next four lines to remove empty document from DTM to prevent potential errors.
#rowTotals<-apply(dtm,1,sum) #running this line takes time
#empty.rows<-dtm[rowTotals==0,]$dimnames[1][[1]]
#corpus<-corpus[-as.numeric(empty.rows)]
#dtm <- DocumentTermMatrix(corpus)

inspect(dtm[1:5, 1:5])
findFreqTerms(dtm, 20)
dtm.mx <- as.matrix(dtm)

#calculate term frequency
frequency <- colSums(dtm.mx)
frequency <- sort(frequency, decreasing=TRUE)
frequency[0:24]
```

The R tool iterates the topic modelling for all the words present and generates the term-document frequency matrix. This gives us the top topics for our corpus. It also generates Document term matrix.

The word cloud generated from tdm can be seen as:



The R tool gives us the topics and corresponding subtopics. It also maps each document to a corresponding topic. We take this data and modify our earlier json file of the collected tweets.

For mapping each tweet to a topic, we created a field "tweet_topic" and mapped the LDA data to that field using a java program. This program read the json file, created the new field and populated it with the corresponding topic id as per LDA returned data.
3. Solr Backend:
The newly generated json file after LDA modelling with added attributes was then posted to solr. Solr then generated an index for this updated twitter data. The entire further processing and querying for topic summarization was performed on this indexed solr data.

4. Designing the User Interface:

4.1 The home Page:

The index file which contains the main page of our UI is as shown below. This web page contains the Search box through which a user can query the data which would be processed and relevant results would be provided later. It also contains our logo (as promised in the demo presentation).
Our project is live and one can access it at our url hosted on AWS ubuntu instance.
http://54.191.105.98:8983/solr/IR4/index.html

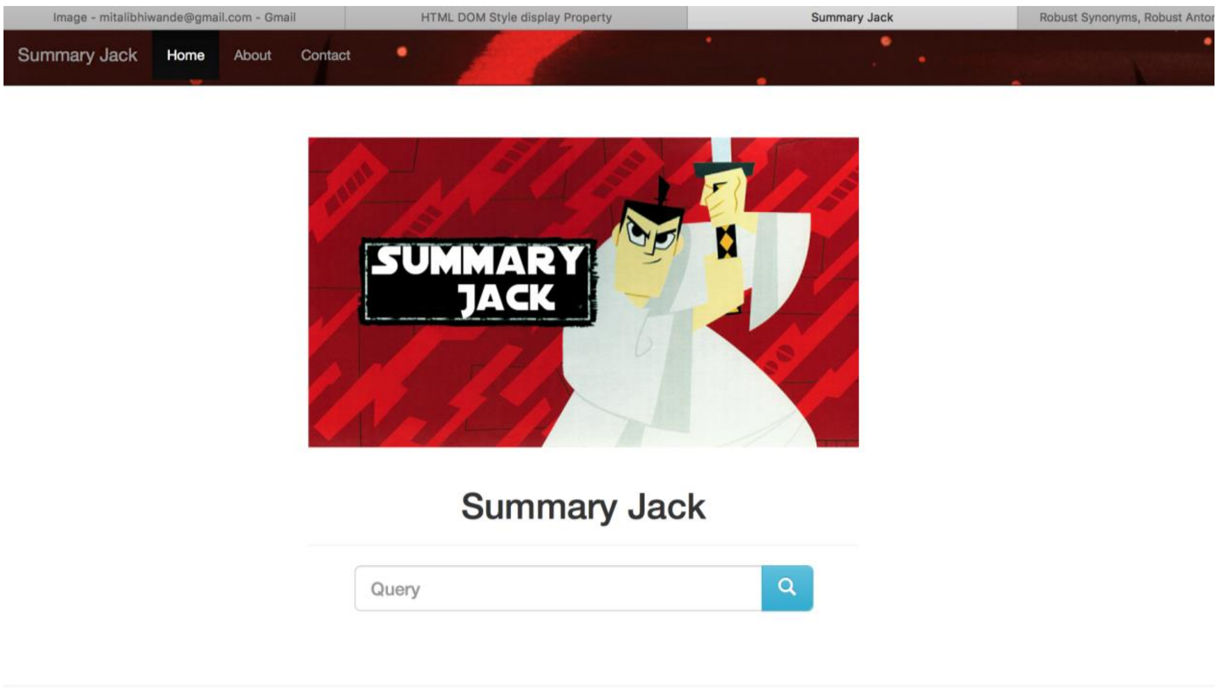
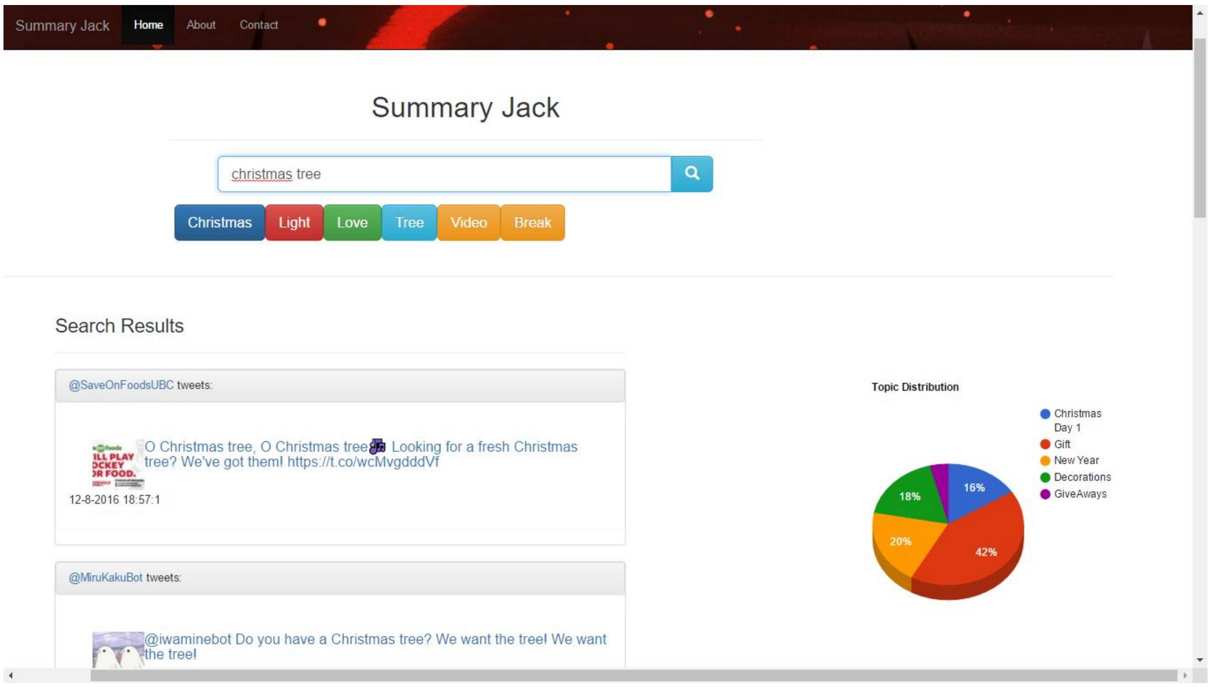


Image - mitalibhiwande@gmail.com - Gmail
HTML DOM Style display Property
Summary Jack
Robust Synonyms, Robust Anton
Summary Jack
Home
About
Contact
SUMMARY JACK
Summary Jack
Query

4.2 On Query:

If a user queries “Christmas tree”, the following results will be displayed on the screen. The various components of this page are:



@SaveOnFoodsUBC tweets:

O Christmas tree, O Christmas tree Looking for a fresh Christmas tree? We've got them! https://t.co/wcMvgdddVf

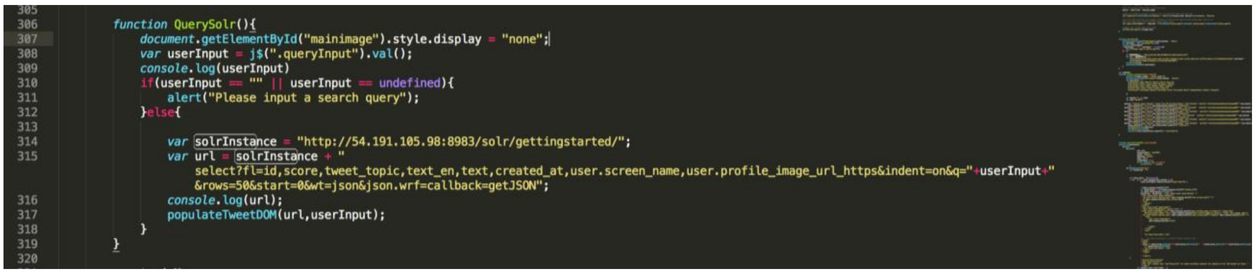
12-8-2016 18:57:1

@MiruKakuBot tweets:

@iwaminebot Do you have a Christmas tree? We want the tree! We want the tree!

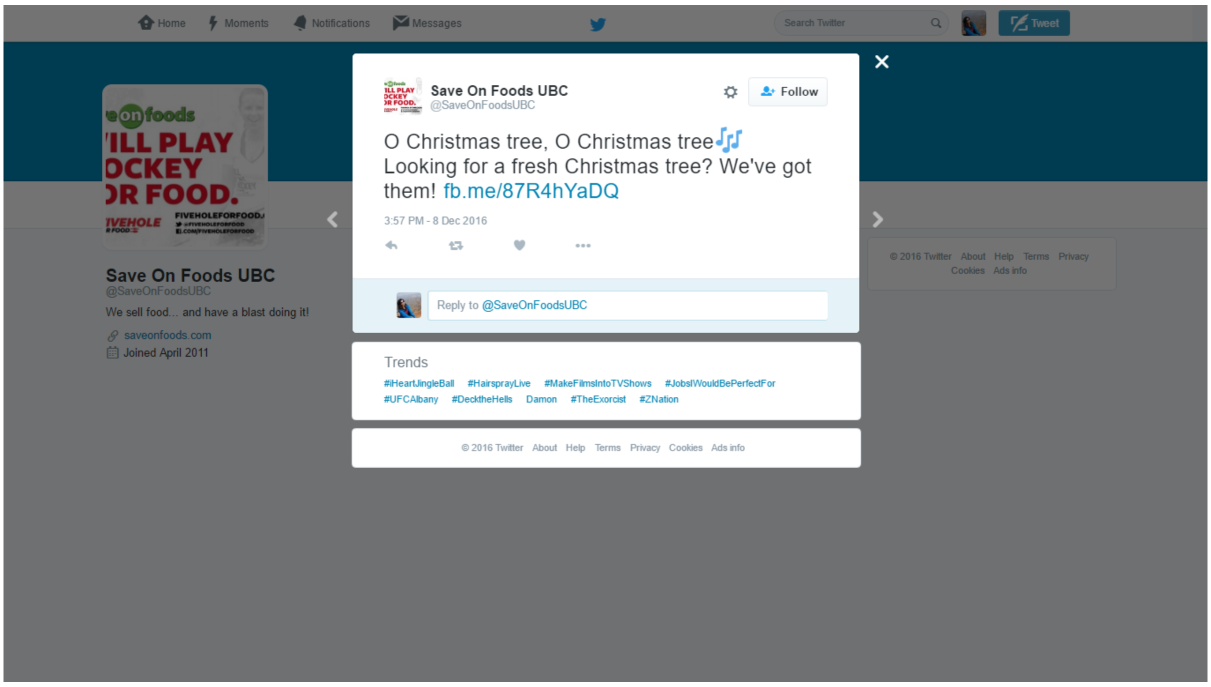
4.2.1 Top tweets:

Our system also returns the top tweets for that particular query.
The obtained tweets are dynamically queried to the solr instance using the query function:



```
function QuerySolr(){
    document.getElementById("mainimage").style.display = "none";
    var userInput = j$(".queryInput").val();
    console.log(userInput)
    if(userInput == "" || userInput == undefined){
        alert("Please input a search query");
    }else{

        var solrInstance = "http://54.191.105.98:8983/solr/gettingstarted/";
        var url = solrInstance + "
            select?fl=id,score,tweet_topic,text_en,text,created_at,user.screen_name,user.profile_image_url_https&indent=on&q="+userInput+"
            &rows=50&start=0&wt=json&json.wrf=callback=getJSON";
        console.log(url);
        populateTweetDOM(url,userInput);
    }
}
```

These tweets are displayed in panels one below the other with the date and time they were created. (as displayed above) If a user clicks on any particular tweet, then the user is directed to the twitter profile of that particular user as shown.



Save On Foods UBC
@SaveOnFoodsUBC

O Christmas tree, O Christmas tree
Looking for a fresh Christmas tree? We've got them! fb.me/87R4hYaDQ

3:57 PM - 8 Dec 2016

Reply to @SaveOnFoodsUBC

Save On Foods UBC
@SaveOnFoodsUBC
We sell food... and have a blast doing it!
saveonfoods.com
Joined April 2011

Trends
#iHeartJingleBall #HairsprayLive #MakeFilmsIntoTVShows #JobsIWouldBePerfectFor #UFCAlbany #DecktheHells Damon #TheExorcist #ZNation

© 2016 Twitter About Help Terms Privacy Cookies Ads info

4.2.2 Topic Distribution:

A 3D pie chart representing the main topic distribution is displayed. These are the main topics provided by LDA depending upon their similarity scores. The pie chart is developed using google charts. It shows a distribution of total number of tweets assigned to a particular main topic. The pie chart changes its topic-wise distribution dynamically on different queries the user runs on the system.

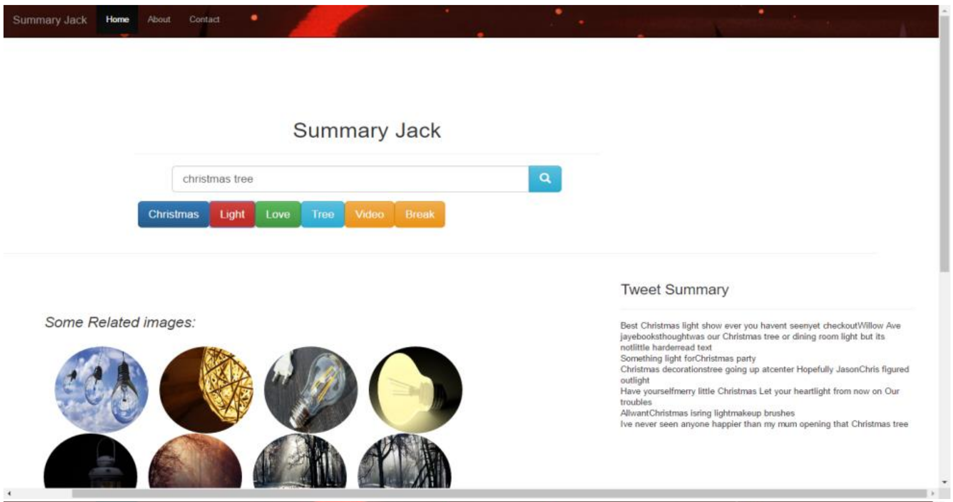
4.2.3 LDA modelled subtopics:

The resulting subtopics from LDA are displayed below the search bar.
A user can click on them and obtain the relevant summary. These subtopics are rendered dynamically based on the count of topic for the retrieved tweets by solr.

Since we already have indexed the tweets mapped to a specific topic, we maintain a count for each topic. After the results the returns by solr, have the total topic counts for all tweets. Then we decide the main topic as the one with highest frequency for returned tweets. And the corresponding subtopics for that topic are displayed as buttons on the interface. The Subtopics for the topics are the ones modelled by LDA.

4.2.4 Summaries:

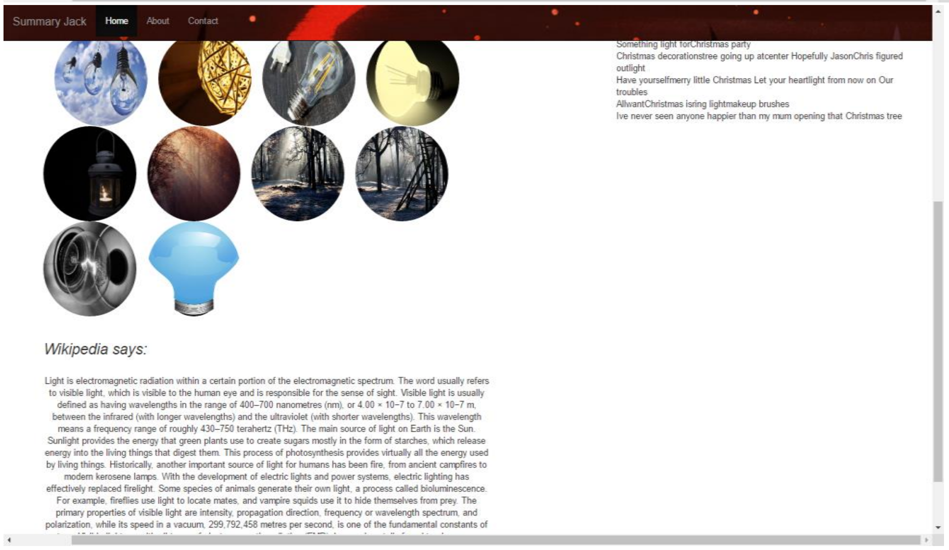
When the user clicks on a sub topic, its corresponding summaries are displayed as follows:



Some Related images:

Tweet Summary

Best Christmas light show ever you havent seenyet checkoutWillow Ave jayebooksthoughtwas our Christmas tree or dining room light but its notlittle harderread text
Something light forChristmas party
Christmas decorationstree going up atcenter Hopefully JasonChris figured outlight
Have yourselfmerry little Christmas Let your heartlight from now on Our troubles
AllwantChristmas isring lightmakeup brushes
Ive never seen anyone happier than my mum opening that Christmas tree



Something light forChristmas party
Christmas decorationstree going up atcenter Hopefully JasonChris figured outlight
Have yourselfmerry little Christmas Let your heartlight from now on Our troubles
AllwantChristmas isring lightmakeup brushes
Ive never seen anyone happier than my mum opening that Christmas tree

Wikipedia says:

Light is electromagnetic radiation within a certain portion of the electromagnetic spectrum. The word usually refers to visible light, which is visible to the human eye and is responsible for the sense of sight. Visible light is usually defined as having wavelengths in the range of 400–700 nanometres (nm), or 4.00 × 10−7 to 7.00 × 10−7 m, between the infrared (with longer wavelengths) and the ultraviolet (with shorter wavelengths). This wavelength means a frequency range of roughly 430–750 terahertz (THz). The main source of light on Earth is the Sun. Sunlight provides the energy that green plants use to create sugars mostly in the form of starches, which release energy into the living things that digest them. This process of photosynthesis provides virtually all the energy used by living things. Historically, another important source of light for humans has been fire, from ancient campfires to modern kerosene lamps. With the development of electric lights and power systems, electric lighting has effectively replaced firelight. Some species of animals generate their own light, a process called bioluminescence. For example, fireflies use light to locate mates, and vampire squids use it to hide themselves from prey. The primary properties of visible light are intensity, propagation direction, frequency or wavelength spectrum, and polarization, while its speed in a vacuum, 299,792,458 metres per second, is one of the fundamental constants of

5. Generating Summaries:

On clicking the on the subtopics, user sees the summarization for that subtopic.
For the summarization part we decided to provide three types of summaries:

1. Based on Wikipedia data
2. Images summary
3. Tweet Summary

- **Wikipedia Summary:**

Here we have used Wikipedia API to retrieve the first paragraph of the clicked subtopic, it queries the API and returns a json file which is formatted as plaintext i.e. the hyperlinks are removed and text is formatted to give paragraphs. URL is as follows:

"https://en.wikipedia.org/w/api.php?format=json&action=query&prop=extracts&exintro=&explaintext=&titles="+ subtopic;"
Where subtopic is loaded dynamically on users click data.

- **Image Summary:**

For this we tried out various API's like Bing, Flickr, and Google's custom search. We decided to use Pixabay API for image retrieval as it allowed a limit of 5000 images per hour and provided copyright free images.
When queried it returns a json file with various fields such as urls for different sizes, we can choose the size we want. The URL is as follows:
"https://pixabay.com/api/?key=**q=Christmas+"+subtopic+"&image_type=photo&pretty=true";

- **Tweet Summary:**

This is based on the tweets present under that subtopic. It loads dynamically on choosing different topics. We decided to choose the retweet count parameter for populating the same as it suggests a high relevancy for tweets with larger retweets. The tweets that belong to the clicked subtopic, and present in the respective topic that are relevant are displayed as tweet summary. Since the corpus is small as of now, we have set the minimum required retweet count as 100.

6. Team Members:

- **Setting up Solr and LDA modelling:** Tariq Siddiqui
- **UI functionalities**: Junaid Shaikh, Tariq Siddiqui
- **UI Designing:** Mitali Bhiwande, Tejasvi Sankhe, Junaid Shaikh
- **Report:** Mitali Bhiwande, Tejasvi Sankhe

7. Discussion:

- Implemented subtopics using LDA model and tuned it for optimum results.
- UI loads subtopics dynamically based on returned results topic distribution.
- Summaries are rendered dynamically.
- Provides three kinds of summaries. (Wikipedia summary can be changed to provide summaries from other sources as enhancement).

8. Updates(for next iteration):

- Twitter data is limited for the system
- LDA modelling needs to be done again if new data is collected(can be automated)
- Dynamically generate tweet summary using NLP techniques.

9. References:

1. LDA:

- RStudio
- http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/
- Empirical Study of Topic Modeling in Twitter,

  http://snap.stanford.edu/soma2010/papers/soma2010_12.pdf

2. API used

- Wikipedia
- Pixabay
- Google Charts