

Extracting Topic Trends and Connections: Semantic Analysis and Topic Linking in Twitter and Wikipedia Datasets.

Ivan Marcin(ivmarcin@stanford.edu)

Sam Shiu (bwshiu@stanford.edu)

CS224W Final Project, Group 4, Autumn 2012, Stanford University

1. ABSTRACT

In the past, research has proposed that in order to process language in articles or human communications, computers require access to vast amounts of common sense and domain specific knowledge, just similar to what human would need to understand the content of articles to provide a topic or a summary. With the advent of Wikipedia, we find now a corpus of data where one can better classify a text or an article into a topic or mixed topics. We assume that pages on Wikipedia are very highly related to individual topics, with a topic being a thing, an event, a person, etc. We also assume that the level of interest in people's conversations would relate to the level of interest for a topic, and we can connect topics based on whether the same group of people talk about the same topics, so if a golfer group collectively talk about both birdies and sand traps, we can assume there is a connection between those 2 topics.

The contribution of this paper is threefold. First, we used word distributions to represent semantics of Tweets grouped in certain time frame, and we compute the semantic relatedness of Tweets to Wiki topics. Second, we used topic correlation to generate topic relatedness over time. Finally, we established the topic trend connection with the amount of Tweet activity topics over a span of 6 months and validate the accuracy of the generated trends by comparing it to search volume data from Google Trends.

2. INTRODUCTION

How do humans summarize an article? We would say a human summarizes the specific wording of an article, based on their background knowledge and experience. In contrast, computers will have difficulty to interpret the contents of articles or human communications such as micro-blogs or tweets. One commonly explore idea is that of relating topics together simulating the human process, by trying to understand the meaning and relations of information. Computers would require access to, and process vast amounts of data, and deduction of context, common sense, and domain specific knowledge is an extremely large problem to tackle. Apart of the processing, we would need a way to let the machine learn, in a similar way to what a person would need in terms of education, background, speech, etc. to understand information such as the contents of articles, and interpret or summarize them into topics or categories.

With the advent of Wikipedia, a collection of human knowledge, a world of understanding is presented in a series of semantic representations. Instead of trying to connect information through the meaning, we propose the use of massive datasets of both knowledge (Wikipedia), and human conversations (Twitter) as a mean to extract these relationships. Since any article can be composed of mixed topics or a combination of several topics, any article or chunk of information can be represented by a group of Wikipedia topics, which would relate to a subset of typical human articles or communications. Thus, one can classify a text or article into a topic or mixed topics by the relatedness of words that are normally distributed in any Wiki topic. First, we assume that a word distribution for a given topic is somewhat unique or close to unique for each topic, and closely related topics would have similar word distributions. These word distributions then would appear in a similar manner in people's conversations, and matching them both would direct us to relate a topic, to a conversation. However, even though we relate word usage in this paper, we do not distinguish other language characteristics other than word usage, such as the sequence of words in this paper or ambiguity of words. Thus, we used word distributions to represent semantics of Tweets grouped in certain time frame, and to compute semantic relatedness of Tweets to Wiki topics by using Explicit Semantic Analysis (ESA). Second, based on the connection between topics to the tweets, we generated network graph of topics over time, with each topic consisting of a topic and a representation of the tweets that are linked together to that topic, and connect nodes or topics through edges calculated by how many items the twits of a topic have in common, such as users, or hash tags. We used this topic correlation to generate topic relatedness through community detection. Finally, we collected a sample of data from Google Trends using the same topics we mined from Wikipedia, i.e. the search volume of such key words or topic words in that period to verify our relatedness results. The results we obtained from twitter conversations are strikingly similar on how a topic trends up and down over time in search volume. We established this high level of Google trend connection with a set of Tweets topics over a span of 6 months.

3. PROBLEM FORMULATION

3.1.1. Overview of the problem

We wanted to answer the question of what is being relevant at a given time, and based on that relevance, where would that relevance lead us to. In real life, events and situations aren't isolated and things like a blizzard trend at a given time and are connected to other events like winter and storms. Our theory is that not only topics bubble up and down in relevance from time to time, but connections between this relevant event bubble up and down as well.

Summarizing the above, we want to know what is trending and what led to it. To do so; we need to be able to find good topics, calculate what their trends are, and find how they are connected to each other.

3.2. Itemized Problem Statements

3.2.1. Generating Relevant Topics to Track

First we need to know what to look for. For this we need to know what a "topic" is, irrelevant of whether is a common thing or a new topic or event happening. We also need to know what to be looking for based without human intervention and to be able to find out new topics that here previously unknown. We took a look at auto generation of topic techniques like bags of words, LSA, or looking for human generated trends like hash-tags, and our approach was to mine topics from Wikipedia. Topics are cleanly and carefully curated by humans and the update/creation of new content speed rivals that of other sources like news and blogs.

3.2.2. Generating trend historical data based on people chatter

Next we need to know the relevance of each topic at a given time, and how measure how it changes over time. We had the alternatives looking at search trends and page views for each topic on Wikipedia. Search trends are grouped by keyword rather than topic making both association and access another problem (which keywords to track for which topics, and how to group each trend result). Another alternative was to use Wikipedia view counts over time, though we empirically know that a person may consult a topic a couple times but their chatter about it is spread through longer periods of intermittent time. Based on this we thought that tracking how people talk about a

topic on twitter would better represent this, so we need a way to correlate tweets, to Wikipedia topics.

3.2.3. Connecting Topics together based on how people talk about it.

Events are not isolated, the rise of a musician will happen because of a concert or a song, the sudden increase importance of a Country would happens due to an event. Our theory is connections between topics are neither fixed nor static, and the degree of by how they're connected changes over time. Rain could be very connected to the Korean actor "Rain" on some time, and more connected to the eastern monsoon on others. We needed a machine-generated way to identify these connections and link topics together. Our problem was how to mine connections from either Wikipedia or twitter conversations.

4. PROPOSED METHOD

4.1. Overview of the model

We propose a way to mine topics and their relevance by extracting human curated topics from Wikipedia, and then performing semantic analysis over both the page content for those topics and twitter data to correlate chatter to topics. With the generated values of chatter amount per topic we can then generate a directed graph based on what topics have in common on their related conversations on twitter. Clustering that topic graph would allow to groups of topics related by their usage. The overview of the system proposed is as depicted in figure 1.

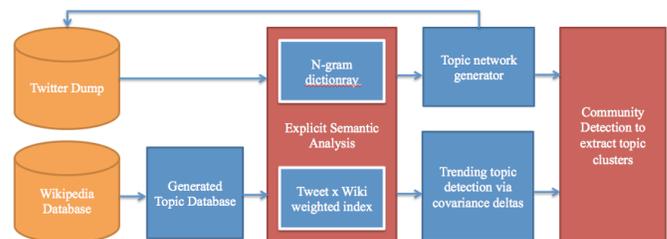


fig 1. Architecture diagram of the system.

4.2. Topic Formulation based on Wikipedia.

Wikipedia is an excellent source of topics since information is neatly curated into a one topic per page, including individual pages for each meaning of ambiguous keywords. Another interesting concept is that subjects, people, events, places, companies, etc.

are each packed into a single page closely defining a topic. Given this structure we went through the simplest approach of making each title of a disambiguated page from Wikipedia an individual topic for our program.

4.3. Data Dumps and indexing.

Corpus of Wikipedia metadata comes from a database dump collected from the Wikimedia archive. It consists of all the pages, categories, and links until October of 2012, with approximately 28.74 million pages and 69.93 million category links. Topics then were indexed and ranked by their view count, and to simplify data analysis, we chose the top 15,000 articles based on ranking to be crawled, extracting the actual page content for the topic.

Our corpus of tweets comes from Stanford SNAP's dataset collection by Jure Leskovec [1]. It consists of approximately 476 million tweets (although we did not use the entire set) from June 11, 2009 to December 31, 2009. We divided the data into separate days, and used approximately 1-2 million tweets per day (except for July 15, 30, 31, and October 31, for which there was no data). There is a total of 201 days of tweet data.

4.4. Lexical analysis and collection of n-gram dictionary

We collected a dictionary of words to match tweets and wiki pages with, combining n-gram dictionaries from 3 different sources:

1. Alex Davies [2]: Specifically created for analyzing Twitter, this list consists of approximately 7400 words. We felt the fact that it included very common words was important in making the most use of our data.
2. Fiction: A list of approximately 2000 of the most commonly used words in fiction, obtained from Wikipedia.
3. Words of human state of mind: A list of about 4153 of most used in expression for human state of mind. This list is generated from 65 seed words of the POMS [3], into a list of 4153 words from the Google n-grams dataset [4], [5].

This resulted in a lexicon of approximately 12630 words or tokens. In order to reduce most commonly used words, Blei [6] suggested to use 90% of documents as a cut off of coexisting words, we further reduce it to 12,023 words by taking out the words that cover 70 percent of all documents, i.e. tweets or Wiki dataset.

We proceeded to generate 2 matrices with the count of n-grams found in each dataset, one with counts for each wiki page, and other for counts on tweets by day. Removing articles from the top 15K topics which where disambiguation, were stubs/redirects, or had less than 100 words, we were left with 14,328 topic models as suggested in [7]. Thus, we generated two matrices; a tweet matrix of size 12,023 n-gram tokens by 201 days, and a wiki matrix of 12,023 n-gram tokens by 14,328 topics.

The values of the tweet matrix were normalized corresponding to day m and word i , since the tweet sample count differed for each day, and this was calculated as:

$$p_i^m = \frac{\# \text{ of times word } i \text{ appeared on day } m}{\sqrt{\sum_k (\# \text{ of times word } k \text{ appeared on day } m)^2}}$$

The wiki matrix was normalized by corresponding topic n to words i , to adjust for length of page content and calculated as:

$$q_i^n = \frac{\# \text{ of times word } j \text{ appeared on topic } n}{\sqrt{\sum_k (\# \text{ of times word } k \text{ appeared on topic } n)^2}}$$

4.5. Generating relevance using Explicit semantic Analysis to link tweets to wiki pages.

In order to connect the matrices generated for both Wikipedia and twitter, we performed a method similar to ESA [13], we simply use a simple method where the following equation was used:

$$\begin{aligned} p(\text{topic}_n | \text{day}_m) &= \sum_{k=1}^i p(\text{word}_k | \text{day}_m) * p(\text{word}_k | \text{topic}_n) \\ &= \sum_{k=1}^i p_i^m * q_i^n \end{aligned}$$

where i is the total number of words in the lexicon space that we are analyzing. This will generate a new matrix of *Tweet Days* \times *Topics* where the value for each is the lexical match between both datasets. The word count generated for each value indicates how much chatter was in twitter for a given topic on each day.

4.6. Connecting topics by how measuring commonalities in twitter chatter among related topics

We initially tried the approach of using the LDA algorithm [12], in particular we used fast LDA [13] to generate both 50 groups and 100 groups of the 14,328 topics.

We also used a Singular Value Decomposition SVD method to extract 100 grouping of the same number of topics.

We found those data are highly correlated with about every other grouping of data. One explanation is that

the tweets dataset could be too large for such analysis; the other reason is that the 100 groups of 14,328 topics are relatively insufficient to generate adequate meaningful groupings. One way to improve such analysis is to increase the tweet division by hourly or every 15 minutes as a single document.

Given we had the list of top X topics for a given day, we proceeded to relate them by constructing a network of our model consisting of nodes representing a “Mega-tweet” for each topic. We generated one node per topic and assigned a node weight based on the number of tweets that lexically match the content of the topic on Wikipedia. This value is the one extracted on the ESA matrix. Edges between nodes on the network were generated by calculating what tweets were common to both nodes on the network using the following model:

$$CommonTweets(n,m) = \{Tn \in V\} \cup \{Tm \in V\}$$

$$dct(n,m) = \sum_n \frac{1}{n} Tn \in V + \sum_m \frac{1}{m} Tm \in V$$

Given that keeping track of every single tweet for every topic node and doing a match between against every other node is exponential in both time and space, making it close to impossible to calculate in finite time given the sizes of the datasets. As an alternative we proposed a random sampling algorithm that calculates a sample size of random tweets for each node based on the ESA calculated weight of a node, and match the tweets by looking up how many users and hash tags tweets from node N are in common with node M. This greatly simplifies the memory requirements and empirically, ~ 500 random tweets per node are required to generate meaningful connections.

The formula to generate the sample size is the following:

$$SampleSize(n) = \frac{weight(n) \cdot \sum_{i=1}^n weight(n_i)}{TweetPoolSampleCount \cdot samplerate}$$

This way instead of equally sampling for example 1% of the complete tweet pool for every node, we vary the sample size based on importance of the node and the range weights across the graph.

Algorithm : Random Tweet sampling and linking through common users and hash tags.

Initialize graph **G** with the top X nodes sampled from ESA.

Initialize a dictionary **TopicData** holding the contents of Wikipedia page for each topic in **G**

Initialize a list of the most common English words **CommonWords**

```

1: For each Node N in G do
2:   Crawl Wikipedia, retrieve the page content for topic N
3: End for
4: For each node in graph G do
5:   Compute sample size of tweets St for N
6:   Do
7:     Retrieve random tweet from dataset by randomly accessing a
       point in the dump file and retrieving tweet at that location.
8:     Remove words existing in CommonWords from tweet
9:     For each word in tweet do
10:      Count occurrences of word in TopicData{N}
11:      If occurrences > matching threshold
12:        Add tweet to node
13:     End for
14:   Until tweet count = sample Size N
15:   For each node M in G
16:     For each node M in G
17:       Extract usernames and hash tags in N
18:       Extract usernames and hash tags in M
19:       Count users and hashes in common
       between N and M as common items Ci
20:       If {Ci/Weight(N)} + {Ci/Weight(M)} > threshold
21:         Create edge with weight {Ci/w(n)} + {Ci/w(m)}
22:     End for
23:   End for

```

We calculate indicate how large was the match by calculating the common matches through:

$$Cm = \left[\frac{Ci}{Weight(N)} \right] + \left[\frac{Ci}{Weight(M)} \right]$$

Where Cm are the count of usernames and hash tags in the tweet samples for nodes N and M. We do this based on the idea that how well 2 nodes are connected is based on the percentage of matches given the node size. For example a small node with a weight and tweet sample size of 100 with 90 of those being matches to another node mean a very strong connection, while 90 matches on a node of sample size one thousand would mean a very weak connection.

4.7. Clustering by measuring commonalities between related chatter for each topic

Once we created a graph with the topics and connections, we approached the detection of linked nodes by performing a grouping via community detection using networkX and mathematica, following the implementation of the modularity maximization method. We tuned the community detection by adjusting the number of edges created by testing different thresholds for the values used in our edge creation algorithm, specifically the following thresholds:

Ts = Size of the twitter sample selected per node
C = Min value of Cm

T_m = How many lexical matches were found between a tweet and a wikipedia page

We identified that tweet samples per node of under 200 performed very poorly and over 10,000 increased computational time exponentially. We found that an average of around 1500 tweet samples per node to be the sweet spot.

Lex matches between tweets and wiki pages on the sampling had to be kept at a minimum of 5. Lowering the matching would select tweets too random and disconnected to the topics to create any meaning and too high increased computational time trying to find tweets in the tweet dump. Also a common item match threshold was kept from 70 to 85% percentage. Above 85% too little edges were generated and under 70% the network connectivity was too high for the community detection algorithm to detect meaning.

5. RESULTS

5.1. Trending topic validation

Based on the ESA matrix generated we were able to graph trending topics for a given day. Given our data showed spikes of interest when the matching was graphed over time, our apparatus to validating the results became a comparison of our trending data with Google's search volume trend data. By searching in Google for topics that in our matrix generated a high variance in a short period of time we were able to find the top topics for a given day, and we validated the accuracy of the results by plotting both our trend and Google trend data.

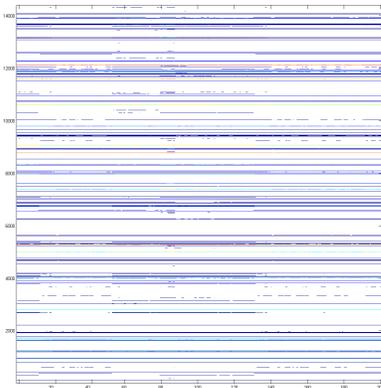


fig 0 – Plot of relevance per topic over time. X axis indicates time and the intensity of the line indicates relevance for a topic on tweets.

We selected 150 topics which showed interestingly significant results and whose data within each topic demonstrated over 3σ variance, or spikes of the each day's data for this comparison.

Figure 2 and 4 show the topics “Patrick” and “Snow” demonstrating the accuracy of this method. Fig 3 shows the values taken directly from a screenshot of the Google trends website for the word “Patrick”. In the graphs, the blue line shows the Google Trend search from the period of 6/11/2009 to 12/31/2009. The red line shows the $p(\text{topic}_n | \text{day}_m)$, where topic is snow for the same period. One can explain that people start to expect snow at the middle of the autumn, and fully discuss the snow topic snow in the midst of the winter season. The correlation, which shows 0.70 indicates the higher correlation value.

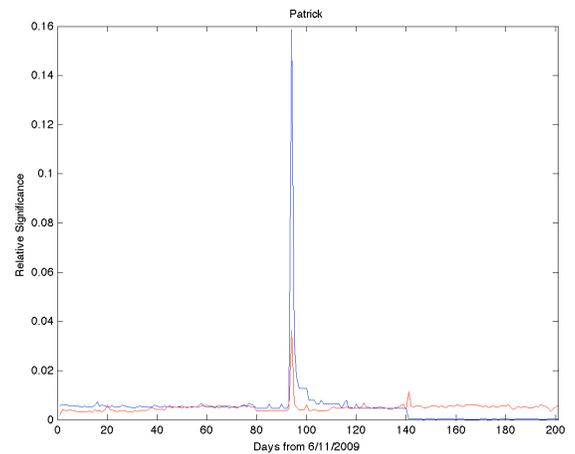


fig 2 - Results for twitter chatter for topic “Patrick”. Red: Our method. Blue: Google trends Data

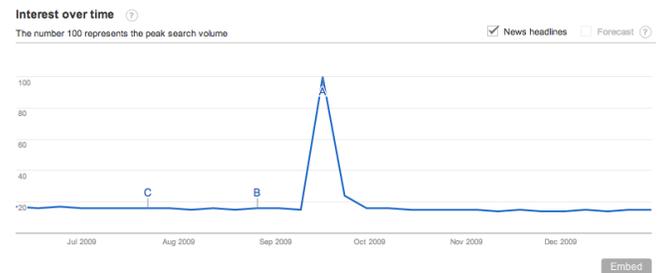


fig 3- Google trends screenshot of search volume for “Patrick”

The same increased chatter spikes were found for other hot topics like Christmas, Snow around December. Seasonal items like “Panetone” popped around December as well as “Turkey (bird)” around November, the time for Thanksgiving where turkey is quite popular. Other interesting spike volumes are around personality names like “West Flemming”.

Topics like “Nobel Prize” had a spike around the time where the prize is awarded. All of this spikes where correlated and validated with search volume data from Google trends.

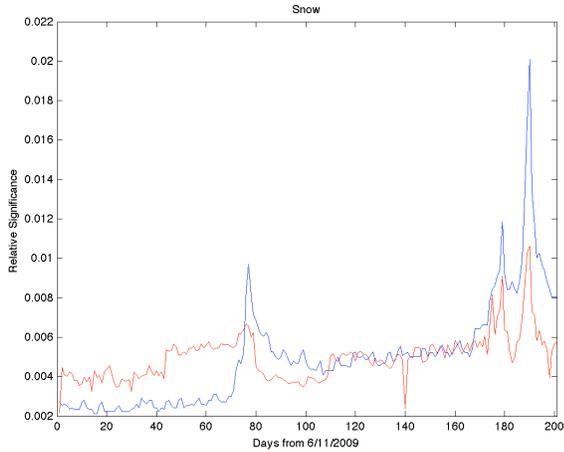


fig 4- Trend plot for the word “Snow”- Red: Our method, Blue: Google Trends

The following table shows the correlation coefficient of each matched topic from the above method. About 50% of the Google Trend shows a high positive correlation with the tweet-wiki generated topic. Although the correlation does not show a high value in less than 50% of the case, we believe that the exact match of the Google search trends can’t be matched due to a large population of search, and the fact that some more generic search keywords apply to different topics. The second reason is that the variability of people search in Google may have different intension than the search strings can demonstrate. Nevertheless, we believe the confidence level of such coincident is high.

Topics	Correlation Coefficient
John Nash	0.03
Mister Magoo's Christmas Carol	0.66
List of meteorological phenomena	0.09
Declaration of independence	0.34
Patrick	0.84
Michael Faraday	-0.03
Christmas tree	0.58
Ballistic missile	-0.29
Christmas	0.83
Francis Pharcellus Church	0.66
West Point (disambiguation)	0.10
Nobel Prize in Physics	0.46
Panettone	0.62
Snow	0.70
List of football clubs in Spain	0.03
Yule	0.64
Turkey (bird)	0.63
Holiday	0.57
Cornucopia	0.08
November	0.31
The Gift of the Magi	0.57
West Flemish	-0.14

Table 1- Correlation coefficient of a sample of topics. Used to detect topics with “spikes of interest”

5.2. Information grouping through community detection

Once we had a definition for nodes and their relevance at a given time, the next step was to generate connection between them utilizing a grouping technique. Our first approach was to do a topic grouping by using SVD to find groups of overlapping n-grams. Although it was possible to find groups based on language similarity this didn’t led to groups of very high value in terms on meaning.

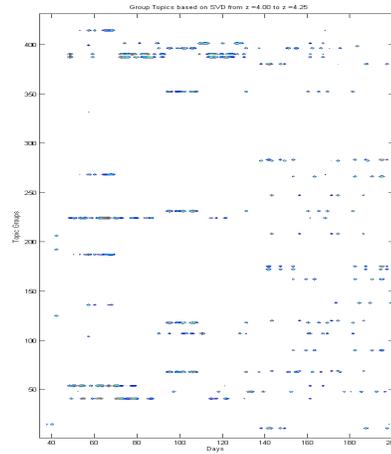


fig 5a- plot of topic grouping using the SVD methodology

For our approach alternative to generate connections between topics, we proceeded to do a graph generation method, in conjunction with a community detection algorithm for the topics that showed to be trending from our previous results. We proved the method by generating the graph multiple times validating that on each iterations the same connections were generated despite of selecting different samples of tweets, and from different days. We also performed a test of the graph generation method with random data, generating random edges and random weights as a comparison point. We found that topics like “Christmas”, “Snow”, “Christmas Carol”, “Christmas Island”, “Christmas tree”, “Pennine Alps” are group together in the same community with different tweet datasets coming from different days of tweets. Each dataset was selected by cutting tweets by day thus limiting the tweet sample selection to a single day.

Graphs from the figures 5 and 6 where generated for the same topics on 2 different tweet samples, one from August 17th of 2009, and one from Dec 24 of 2009. We can see the similarity in the communities and topics like those related to Christmas were matched and included in the community on the top left group.

6. FUTURE WORK

As suggested by authors [9], [6], [8], [10], [11], Wikipedia's page content on a topic hardly changes in structure for existing topics, though new topics pop in Wikipedia relatively fast. Tracking Wikipedia for new articles created would be a good way to find emergent topics of interest appearing in real life. We believe that this version of topic modeling method show promising results in comparing to other methods.

Since the usage of words in Twitter is highly dynamic, it makes sense to improve topic trending by utilizing dynamic adjustments; analyzing tweets along shorter periods of time. Furthermore, analyzing data in real time would allow detection of a trending topic, not only for massive amounts of data as in search volume, but also on smaller spaces like small groups of people or conversations.

The results also indicate that a more generic topic would be easier to discover with a day's tweets, such as holiday during long weekends, Turkey topic during at or near the Thanksgiving weekend, Christmas topics during all Christmas season, Green/Patrick in St Patrick day, etc. or pattern words like Friday, which show spikes of interest every week. Pattern extraction of recurring spikes is also an interesting possibility to detect trends in cyclical topics. Further, if we want to identify time sensitive topics such as election results, Presidential rating, economic data, we would need a dynamic minute-to-minute tweet analysis that would yield meaningful information. The current study does show a potential for such application.

The method of LDA does prove to be a useful method as demonstrated by various authors. There is a draw back, however, the LDA tends to group many topics into a very generic where an identifiable trend is rather limited or harder to find. Improving the accuracy of the grouping algorithm would allow to algorithmically link topics not by keyword but by hard connections.

7. RELATED PREVIOUS WORK

Chang et al [10] validated the relevance of topic models using large-scale human experiment. Probabilistic topic modeling is a popular tool due to its ease of use and its unsupervised analysis of text that generates a latent topic representation.

Hoffman et al [11] has demonstrated an efficient way of fitting streaming documents into 100-topic topic model. Ritter et al [9] demonstrated the analysis of large volume of tweets using unsupervised topic model approach in order to learn the conversation structure to use in construction of a data-driven, conversational agent.

Gabrilovich and Markovitch [7] showed a novel method using Explicit Semantic Analysis (ESA) where the concepts are derived from the wiki knowledge and concept space. This method uses co-occurrence information to estimate the

relatedness of wiki knowledge space to the tweeter conversational space.

Ramage et al [8] uses Labeled LDA to summarized Microblogs into 5 different main categories of topics, namely: substance, style, status, social, and other characteristics.

8. CONCLUSION

We found that Wikipedia pages can be used as a mean of collecting topics as an alternative to generating topics through techniques like bag of words. Also, correlating the amount of tweets that are related to each topic, by matching the word count on both the Wikipedia page content and tweets, generates an accurate depiction of the level of interest on a given topic. We also found that connecting these topics together through their level of common items in their respective twitter chats is a possible way to link topics together. e.g: if the set of people who talk a lot about snow also talk a lot about Christmas at the same time frames, it makes sense that snow and Christmas are related to each other based. Finally we were able to prove the accuracy of our methods through empirical data and closely matched the trend lines generated from tweet interest, to search volume data from Google trends.

9. ACKNOWLEDGMENT

We would like to thank Jure Leskovec for his guidance and help with the graphing concepts, suggestions on algorithms and the idea of building a mega tweet network, as well as Bob West for his help on validating our algorithms and help finding related research on syntactic analysis and Wikipedia knowledge.

10. REFERENCES

- [1] J. Leskovec. Snap.stanford.edu. Stanford Large Network Dataset Collection.
- [2] A Davies. Alexdavies.net. A word list for sentiment analysis on Twitter.
- [3] M Lorr, D McNair, L Droppleman, "ProfileofMoodStates," Multi-Health Systems, Inc.
- [4] D. Lin. Dekang Lins Proximity Based Thesaurus. <http://webdocs.cs.ualberta.ca/~lindek/downloads.htm>
- [5] Google n-Grams. <http://books.google.com/ngrams/datasets>
- [6] David M Blei, "Probabilistic Topic Models," Communications of the ACM, Vol. 55, No.4, April 2012.
- [7] Evgeniy Gabrilovich, Shaul Markovitch, "Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis,"
- [8] Daniel Ramage, Susan Dumais, Dan Liebling, "Characterizing Microblogs with Topic Models," AAAI 2010.
- [9] Alan Ritter, Colin Cherry, Bill Dolan, "Unsupervised Modeling of Twitter Conversations," Microsoft Research
- [10] Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, David M. Blei, "Reading Tea Leaves: How Humans Interpret Topic Models,"
- [11] Matthew D. Hoffman, David M Blei, Francis Bach, "Online Learning for Latent Dirichlet Allocation,"
- [12] David M Blei, Andrew Y. Ng, Michael I Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research 3, p. 993-1022, 2003
- [13] Gabrilovich E., Markovitch S., "Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis.