Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)

Mapping Twitter Conversation Landscapes

Soroush Vosoughi,[*] Prashanth Vijayaraghavan,[*] Ann Yuan, Deb Roy
Massachusetts Institute of Technology
Cambridge, MA, USA
pralav@mit.edu, soroush@mit.edu, annyuan@mit.edu, dkroy@media.mit.edu

Abstract

While the most ambitious polls are based on standardized interviews with a few thousand people, millions are tweeting freely and publicly in their own voices about issues they care about. This data offers a vibrant 24/7 snapshot of people's response to various events and topics. The sheer scale of the data on Twitter allows us to measure in aggregate how the various issues are rising and falling in prominence over time. However, the volume of the data also means that an intelligent tool is required to allow the users to make sense of the data. To this end, we built a novel, interactive web-based tool for mapping the conversation landscapes on Twitter. Our system utilizes recent advances in natural language processing and deep neural networks that are robust with respect to the noisy and unconventional nature of tweets, in conjunction with a scalable clustering algorithm an interactive visualization engine to allow users to tap the mine of information that is Twitter. We ran a user study with 40 participants using tweets about the 2016 US presidential election and the summer 2016 Orlando shooting, demonstrating that compared to more conventional methods, our tool can increase the speed and the accuracy with which users can identify and make sense of the various conversation topics on Twitter.

Introduction

Polls have traditionally been the main method through which journalists and public opinion researchers could understand how different issues are playing with the public. However, polls are inherently limited by the questions they contain as pollsters typically ask people a fix set of questions. The relatively recent rise in the use of social media by the general public has made these platforms a potentially more reflective source for understanding public opinion, as the public on social media use their own voices to speak about whatever is on their minds, without being prompted or primed. Twitter specifically, due to its public nature, is an ideal place to get a fresh read on the public. with the public while in social media people use their own voices to speak about whatever is on their minds.

However, as social media have exploded, techniques for measuring public opinion using these platforms haven't kept up. The sheer scale of the data on Twitter presents both opportunities and challenges. On the plus side, this allows us to measure in aggregate how the various issues are rising and falling in prominence over time. But unlike traditional news coverage, data is about numbers. It lacks the human voices and faces that make for compelling stories. Public opinion researchers have been working around this problem by paying attention to the most popular conversation on social media, often identified through hashtags, and anecdotally pulling out citizen comments on those topics. The problem with trending topics is that can overlook non-viral issues that many people care about. And anecdotally selected tweets are not necessarily reflective of the larger conversation.

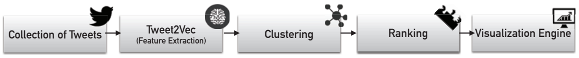


Figure 1: The four-stage pipeline of the system.

Using recent advances in deep neural networks for natural language processing, we developed a tool to automatically identify various clusters of any conversation on Twitter and to identify the tweets that are most characteristic of each cluster. Tweets that are exactly the same, because they were either widely retweeted or sent out repeatedly by a bot, don't qualify. So a characteristic tweet can just as easily come from a person who has very few followers as from a celebrity with millions of followers. It's all about whether the language used is reflective of the broader conversation.

An overview of the system can be seen in Figure 1. As shown in the figure, the system is comprised of four parts: (a) a sophisticated mechanism for extracting rich semantic features from the tweet text, (b) a scalable methodology to agglomerate semantically similar tweets into clusters, (c) a scoring technique to rank the tweets based on how well they exemplify the contents of the clusters and (d) an interactive endpoint to visualize the tweet clusters. Below we explain each of these sections in detail.

Tweet2Vec

Due to the noisy nature of tweets, commonly used methods to extract semantic features like TF-IDF, distributed word vectors (Mikolov et al. 2013), operating at word-level, do not

[*] The first two authors contributed equally to this work.

perform well. Therefore, we utilized Tweet2Vec (Vosoughi, Vijayaraghavan, and Roy 2016), a character-level CNN-LSTM encoder-decoder approach, to learn general purpose vector representation of tweets. These vectors capture abstract semantic structures that can be applied to several generic tasks. Tweet2Vec (Vosoughi, Vijayaraghavan, and Roy 2016) is a recent method for generating general-purpose vector representation of tweets. Tweet2Vec removes the need for expansive feature engineering and can be used to train any standard off-the-shelf classifier (e.g., logistic regression, svm, etc). It uses a CNN-LSTM encoder-decoder model that operates at the character level and can deal with the noise and idiosyncrasies in tweets. Character-level models are great for noisy and unstructured text since they are robust to errors and misspellings in the text. The model learns abstract textual concepts from the character level input of tweets. For example, such models would closely associate the words“n” and “nooo” (both common on twitter), while a word-level model would have difficulties relating the two words. The tweet embeddings generated from this model can help improve the performance of complex linguistic tasks that involve tweets

Training and Evalution

We trained our model on 5 million randomly selected English-language tweets populated using data augmentation techniques, which are useful for controlling generalization error for deep learning models. Data augmentation involved replacing some of the words with their synonyms as mentioned in (Zhang and LeCun 2015; Vosoughi, Vijayaraghavan, and Roy 2016). Similar to Vosoughi et al. (Vosoughi, Vijayaraghavan, and Roy 2016), we evaluated our Tweet2Vec model on a semantic relatedness task, using the SemEval 2015-Task 1: *Paraphrase and Semantic Similarity in Twitter* dataset (Xu, Callison-Burch, and Dolan 2015). Given a pair of tweets, the goal was to predict their semantic equivalence (i.e., if they express the same or very similar meaning), through a binary yes/no judgement. The dataset provided for this task contains 18K tweet pairs for training and 1K pairs for testing, with 35% of these pairs being paraphrases, and 65% non-paraphrases. We achieved performance similar to those reported by Vosoughi et al (F1 score of 0.69).

Clustering

Next, we cluster the tweets based on the tweet embeddings generated by Tweet2Vec (with a vector size of 256) to aggregate semantically similar tweets into a topic bucket. This requires a scalable clustering technique that can take a large number of tweets as input and cluster them in a non-parameterized setting. There are several non-parameterized approaches like bayesian non-parametric models (Hughes and Sudderth 2013). We used a scalable, non-parameterized hierarchical density-based clustering algorithm called Hierarchical DBSCAN (HDBSCAN), introduced by Campello,et.al(Campello, Moulavi, and Sander 2013).

HDBSCAN, , is a clustering algorithm that can be seen as an improvement over existing density-based clustering algorithms. This approach follows Hartigan's model (Hartigan 1975) of density contour clusters/trees and generates a complete density-based clustering hierarchy following the non-parametric model adopted, for an infinite range of density thresholds. As a result, a flat clustering composed only of the most significant clusters based on the stability of clusters can be extracted. HDBSCAN has an input parameter, k, which is a classic smoothing factor in density estimates. The resulting cluster with varying density levels will correspond to different values of the radius ϵ.

Extracting Characteristic Tweets

Next, given a cluster, containing a large number of tweets, we identify the tweets that best characterize the discourse in that cluster. We call those tweets that best exemplifies the cluster as *characteristic tweets*. The characteristic tweets should (a) capture key phrases and words that describes the cluster topic, (b) be self-contained i.e. sufficiently “long” to incorporate some of the key phrases. Graph-based approaches can be employed to rank the tweets based on these criteria. We generate a word-graph from the tweets in the cluster. Formally, let $G = (V, E)$ be a directed graph with the set of vertices V and set of edges E, where E is a subset of $V \times V$. A vertex $v \in V$ represents a word from a tweet in the cluster and a directed edge $(u, v) \in E$ represents the adjacent words where u precedes v. Each edge $(u, v) \in E$ is assigned a weight w based on how frequently the word represented by u precedes the one that v corresponds to. Each vertex in the graph is now scored based on Iterative graph-based ranking.

Iterative graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph; in the context of search engines, it is a way of deciding how important a page is on the Web. Drawing parallels to our system, we employ this technique to score words in the cluster of tweets and eventually rank tweets in a given cluster C. In this model, when one vertex links to another one, it is casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vertex casting the vote determines how important the vote itself is, and this information is also taken into account by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes.

The original PageRank definition for graph-based ranking is assuming unweighted graphs. However, in our model, the graphs contain implicitly devised links, i.e., the edges carry similarity scores, which needs to be accounted for. In this direction we apply a modified version of the Pagerank algorithm introduced by (Mihalcea and Tarau 2004).

For a given vertex V_i, let $In(V_i)$ be the set of vertices that point to it, and let $Out(V_i)$ be the set of edges going out of vertex V_i. The modified PageRank is defined as follows

$$S(V) = (1 - d) + d * \sum_{j \in In(v_i)} \frac{S(V_j) * w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} \quad (1)$$

where d is a damping factor that can be set between 0 and 1.

Starting from arbitrary values assigned to each node in the graph, the computation iterates until convergence below a given threshold is achieved. After running the algorithm, a fast in-place sorting algorithm is applied to the ranked graph vertices to sort them in decreasing order. The modified PageRank can be also applied on undirected graphs, in which case the out-degree of a vertex is equal to the in-degree of the vertex, and convergence is usually achieved after a fewer number of iterations.

Therefore, the modified PageRank assigns a score (pr_{word}) for each of the words in the cluster represented by the vertices of the graph. For each tweet T we calculate a score:

$$score(T) = \sum_{word \in Vocabulary} pr_{word}$$

The tweets are ranked based on these scores and the top ranked tweets are called the characteristic tweets. The highest ranking tweet will satisfy the above mentioned criteria. All the high ranked tweets across various clusters capture diverse views around a given topic.

Visualization Engine

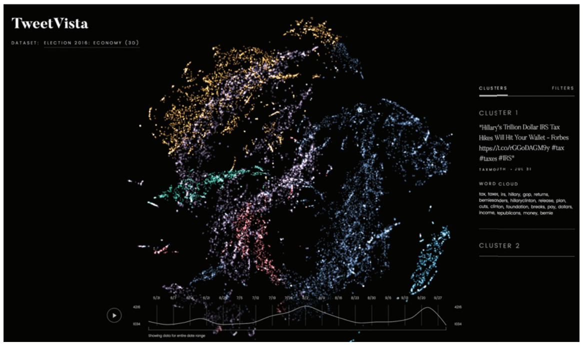
Finally, the visualization engine renders clusters of semantically related tweets as a particle cloud. Users can explore tweets by panning, rotating, or zooming the cloud. Users can filter the tweets shown by properties of their content or authors. The interface also includes details regarding each semantic cluster, such as characteristic tweet and relevant tags within that cluster. Users can choose between several different 2D and 3D datasets to visualize using the tool.

The first step for visualization is the reduction of the high dimensional tweet embeddings to two or three dimensions. We use *t-SNE* for this task (Maaten and Hinton 2008). t-SNE is a dimensionality reduction technique used for this purpose. The technique is a variation of Stochastic Neighbor Embedding (Hinton and Roweis 2002) that is easier to optimize, and produces significantly better visualizations by minimizing the tendency to crowd points together in the center of the map.

Interface Overview

Our tool is a desktop web application best viewed with Google Chrome. Users can upload and visualize datasets containing tweets, after a short processing time the conversation clusters and characteristic tweets are shown. Tweets are represented by particles whose position in 3-D is determined by the t-SNE algorithm described earlier. Users can zoom into the cloud of tweets by using their mouse wheel or trackpad, they can rotate it by dragging along the interface, and they can also pan the cloud's position by pressing their arrow keys. Tweet particles are colored according to the conversation cluster they belong to. Users can see the text, author, and date of each tweet by hovering over it. Users can filter the tweets shown by content properties such as the civility of the tweet (e.g., whether the tweet contains profanity), or properties of the author such as whether the account is verified, the author's number of followers, statuses, followees, etc. Users can also filter the date range from which data is drawn by manipulating the timeline at the bottom of the interface. They can select a particular pre-defined date interval, or press the "play" icon, which allows them to see the clusters forming over time.

The interface includes a side-panel that provides details about each semantic cluster of tweets (such as the most frequently occurring words in the cluster) in the currently visualized dataset.



"Hillary's Trillion Dollar IRS Tax Hikes Will Hit Your Wallet - Forbes https://t.co/rGGoDAGM9y #tax #taxes #IRS"

tax, taxes, irs, hillary, gop, returns, berniesanders, hillaryclinton, release, plan, cuts, clinton, foundation, breaks, pay, dollars, income, republicans, money, bernie

Figure 2: Screenshot of the interactive user interface.

Evaluation

We tested our system on three different datasets:

- Trump's Immigration Speech, 2015
- Orlando Shooting & aftermath, 2016
- Discussion of US Economy on Twitter, Summer 2016

These datasets were collected using a state-of-the-art supervised Twitter topic classifier (Vijayaraghavan, Vosoughi, and Roy 2016). (The details of the topic classifier are out of the scope of this paper, please read the cited paper for more details.) Table 1 shows the characteristic tweets identified for all the clusters in the economy dataset. As it can be seen, an interesting narrative already emerges from these characteristic tweets.

The evaluation is based on user-centric criteria: the accuracy and the speed with which users can navigate and comprehend large number of tweets about an event. To do this, we used the datasets mentioned above. We asked a group of forty undergraduates to examine the datasets. We divided the subjects evenly into four groups. The first group used the complete tool, the second group used our tool minus the characteristic tweets, the third group used a tool that used a conventional topic clustering algorithm (tf-idf plus cosine similarity combined with hierarchical agglomerative clustering), and finally, the fourth group was the control group, their tool did not process the data at all, it put all the tweets into one giant cluster.

Each person worked independently and was allowed ten minutes per dataset. The time limit was set to better differentiate between the different versions of the tool. After the ten minutes had passed, we asked a series of prepared questions about the datasets. For each of the four groups, we averaged the percentage of the questions that they answered correctly. Figure 3 shows the mean, and the standard deviation of each

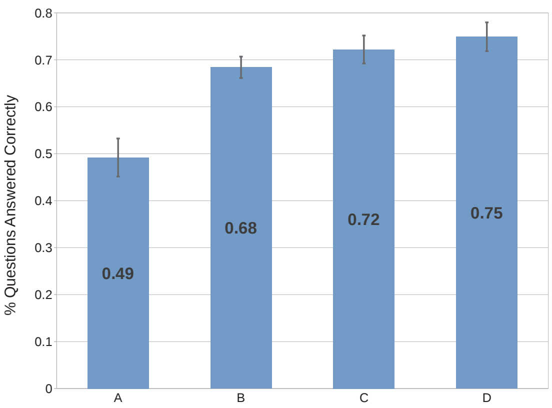


Figure 3: Percentage of the questions answered correctly, using four different tools: A is the control, B is the conventional tool, C is our tool without characteristic tweets, and D is our complete tool.

of the four groups. There was a statistically significant difference between groups as determined by one-way ANOVA ($F(3, 36) = 13.54, p < .0001$). There are three interesting conclusions that one can draw from these results. First, any kind of clustering of tweets significantly increased the performance compared to the control. Second, the addition of characteristic tweets improved the performance of our system by 3%. And finally, the groups using our full system outperformed all other groups.

Conclusion and Future Work

While the most ambitious polls are based on standardized interviews with a few thousand people, millions are tweeting freely and publicly in their own voices about issues they care about. This data offers a vibrant 24/7 snapshot of people's response to various events and topics. In this paper, we presented a tool for mapping the conversation landscapes on Twitter, to better understand what the Twitter public is saying about various issues. We also introduced the concept of characteristic tweets – these are tweets that are reflective of the broader Twitter conversation about an issue – and proposed a method for identifying these tweets. At its core, our tool has a powerful semantic analysis engine that utilizes recent advances in natural language processing using deep neural networks. In contrast to similar tools, our tool was specifically designed to deal with the short, noisy and idiosyncratic nature of tweets. We showed that using our tool, users are able to make sense of large volumes of tweets about in a short amount of time.

An immediate extension to our work would be the automatic labelling of the conversation clusters identified by our system. Moreover, in the future, we wish to extend this work to other social media platforms other than Twitter.

Cluster	Characteristic Tweet - Economy
0	Hillary's Trillion Dollar IRS Tax Hikes Will Hit Your Wallet - Forbes https://t.co/rGGoDAGM9y #tax #taxes #IRS
1	Donald Trump sought cheap labor overseas for clothing lines @CNNPolitics https://t.co/FIAod2zI6O JOB CREATOR: Please bring your jobs back!!!
2	Where Trump, Clinton stand on key economic issues https://t.co/btwowinGtt via @CBSNews #Trump #Clinton #Economy #Economic #Policy #Politics
3	2015 Latino #poverty rate: 21.4%. With HRC more poverty, With Trump more #Jobs, choice is clear. #LatinosWithTrump https://t.co/Ki3hxJYGXt
4	@ZazzyJets Hitler was a progressive. National Socialist. Obama, Hillary, Bush Neo-con GOP are all international socialists.
5	The Best Social Program is a JOB!Trump created JOBS!Hillary created ISIS.#Trump2016 #TrumpTrain #Trump

Table 1: Characteristic tweets of different conversation clusters for the economy datasets.

References

Campello, R. J.; Moulavi, D.; and Sander, J. 2013. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 160–172. Springer.

Hartigan, J. A. 1975. *Clustering algorithms*.

Hinton, G. E., and Roweis, S. T. 2002. Stochastic neighbor embedding. In *Advances in neural information processing systems*, 833–840.

Hughes, M. C., and Sudderth, E. B. 2013. Memoized online variational inference for Dirichlet process mixture models. In *Neural Information Processing Systems (NIPS)*.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research* 9(Nov):2579–2605.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into texts. Association for Computational Linguistics.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Vijayaraghavan, P.; Vosoughi, S.; and Roy, D. 2016. Automatic detection and categorization of election-related tweets. In *Tenth International AAAI Conference on Web and Social Media*.

Vosoughi, S.; Vijayaraghavan, P.; and Roy, D. 2016. Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. ACM.

Xu, W.; Callison-Burch, C.; and Dolan, W. B. 2015. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *SemEval*.

Zhang, X., and LeCun, Y. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.