Using Wordclouds to Navigate and Summarize Twitter Search Results

Rianne Kaptein
Oxyme
Amsterdam, The Netherlands
rianne@oxyme.com

Abstract

This paper describes an application in which wordclouds are used to navigate and summarize Twitter search results. A search on Twitter can return thousands of relevant tweets. By just looking at the first few result pages you will not get an overview of what is discussed in all search results. Our application summarizes sets of tweets into wordclouds, which can be used to get a first idea of the contents of the tweets. Also the application provides the option to zoom in on a certain part of the search results to inspect them in more detail. The application has not been formally evaluated, but we do provide some insights and points for discussion.

1 Introduction

One of the most common problems in Information Retrieval is information overload: there is simply too much relevant information available for the users to process. Therefore applications are needed to help users deal with large amounts of data. In this paper we describe an application which was developed for this purpose. The use of wordclouds in the application serves two purposes:

1. To summarize
2. To aid navigation

This application was developed with the following two user scenarios in mind:

1. General Twitter search
   Nowadays many people express their opinions about products, services and companies on Twitter. When you want to get a broad overview of what people are tweeting in general about a company or event, it does not suffice to read the first few pages of search results. You want to get a feeling for the most frequently discussed topics overall, and dive into particular subtopics of special interest, such as product recommendations.
2. Searching fragments of categorized data
   Besides Twitter there are many more places on the Web where people express their opinions. These opinions can be collected and annotated with labels such as sentiment, source, market etcetera. When you have a large amount of annotated data available, it is interesting to see for example what are the different topics discussed in positive and in negative messages.

In this paper we will focus on the first user scenario: General Twitter search, since Twitter data is abundant and publicly available.

Humans have a great capacity to notice terms which are out of the ordinary. When looking at a wordcloud there will always be some unexpected terms which catch your attention and are good pointers for further investigation. In tweets about public transport you can expect for example tweets about delays, but you might not expect certain tweets about recent events such as a new colour of the trains. What we try to do in the wordclouds is to emphasize the words that are noteworthy from a statistical point of view, and leave it up to the user to decide which messages to explore further.

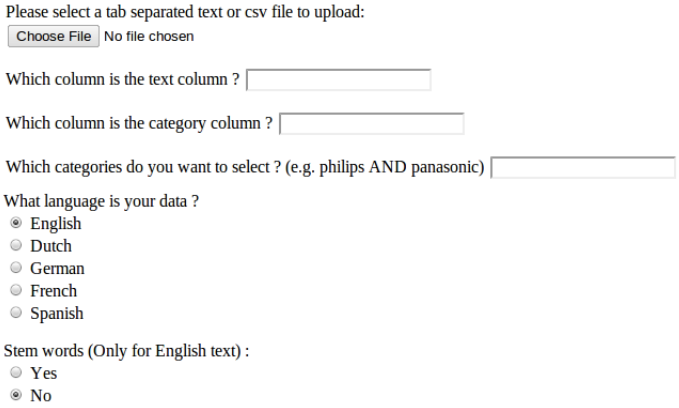
Although the usefulness of tagclouds for navigation is still a topic of debate [2], exploratory applications which make use of wordclouds for summarization and navigation of search results have been moderately successful on specific domains such as web documents [1] and PubMed publications in biomedical literature [5].

The search results that we are investigating in this paper have three characteristics:

- A search result is a short textual message. By design a Twitter message cannot contain more than 140 characters.
- The number of search results is large. If this would not be the case, since the results are short texts, you could simply read through all of them.
- There are many, equally relevant search results. In web search there are usually not more than a handful highly relevant search results. Many of the search results contain copied or redundant information, or only mention the search words occasionally. Although Twitter search results also contain redundant information, i.e. repeated tweets and retweets, the set of relevant tweets can still consist of thousands of equally highly relevant tweets.

In the next sections of this paper we will present our approach (Section 2), a case study (Section 3), and finally our conclusions (Section 4).

Figure 1. First part of the inputscreen



2 Approach

The application consists of two screens. The first screen handles the input, the second screen displays the results based on your input.

On the first screen the system offers a number of selections that can be made to make sure you generate the wordclouds that are best representing your data and your analysis purpose. The input is collected using textfields, radiobuttons and checkboxes. The first part of the inputscreen is shown in Figure 1.

The following selections can be made:

- File selection, a tab separated text file is required as input.
- Text selection, which column in the dataset to use as textual input for the wordcloud generation.
- Category selection, based on a value in any column of your dataset your data can be categorized. It is also possible to create categories based on the presence of words in the contents of your data, e.g. to create a category for all tweets containing the term 'happy'.
- Language, used for the removal of standard stopwords.
- Optionally, additional stopwords can be specified. These words will not occur in any of the wordclouds.
- Stemming, currently available only for English. The Krovetz stemmer is used, because this stemmer always stems words into other valid English words.
- Exclude numbers, when your data includes many numbers such as product prices it can be desirable to exclude these numbers from the wordcloud.
- Exclude retweets / repeated posts, when your data contains a tweet that is retweeted very frequently, this one tweet will dominate the wordcloud which can be undesirable.
- Include only usernames, for Twitter data only, keep only the usernames, i.e. all the words starting with @.
- Include only hashtags, for Twitter data only, i.e. all the words starting with #.

The second screen shows the output, which consists of wordclouds for the categories you have specified, as well as a wordcloud for all the search results.

Wordclouds for categories are generated using a parsimonious language model. This model compares the frequency of words in a set of documents to the average term probability in a background collection containing similar documents to extract the most noteworthy terms. In this case the background collection are all the retrieved search results. Terms that are only mentioned occasionally in the set of documents and terms which have a similar or higher probability of occurrence in the background collection will not be included in the parsimonious language model [4].

The parsimonious language model [3] is an extension to the standard language model based on maximum likelihood estimation, and is created using an Expectation-Maximization algorithm. Maximum likelihood estimation is used to make an initial estimate of the probabilities of words occurring in the set of documents.

$$P_{mle}(t_i|S) = \frac{tf(t_i,S)}{\sum_t tf(t,S)} \tag{1}$$

where S is the set of documents, and $tf(t,S)$ is the text frequency, i.e. the number of occurrences of term t in set of documents S. Subsequently, parsimonious probabilities are estimated using *Expectation-Maximisation*:

$$\begin{aligned} \text{E-step:} \quad & e_t = tf(t,S) \cdot \frac{(1-\lambda)P(t|S)}{(1-\lambda)P(t|S)+\lambda P(t|C)} \\ \text{M-step:} \quad & P_{pars}(t|S) = \frac{e_t}{\sum_t e_t}, \text{ i.e. normalize} \end{aligned} \tag{2}$$

where C is the background collection model. In the initial E-step, maximum likelihood estimates are used for $P(t|S)$. We set the smoothing parameter λ to 0.9. In the M-step the words that receive a probability below a threshold of 0.001 are removed from the model. The iteration process stops after a fixed number of iterations.

In the next section we present a case in which the generated output of the application is presented.

3 Case

Using an example search we will demonstrate how we use wordclouds in our application to navigate and summarize the search results. We executed a search on Twitter using the Twitter search API[1] for the query '#london2012' over the last 5 days, saved all the 30,504 search results in a .csv file and load this file into our application. Looking at the wordcloud over all the results that is shown in Figure 2, we see the term 'torch' is frequently used, and we zoom in on this aspect of the '#london2012' search. By clicking on the word 'torch' a list of messages is shown that all contain the term 'torch', so these messages can be inspected in more detail. This list of messages is still quite long however, consisting of 1,046 tweets. We can zoom in further on these tweets by going back to the input screen and specifying 'torch' as a category. Now, a parsimonious wordcloud is created from the 1,046 tweets that contain the term 'torch'. The resulting wordcloud is shown in Figure 3. The figure is a screenshot of the screen that is displayed when the word 'Sheffield' is clicked, showing the tweets containing the word 'Sheffield'.

Words which occur frequently in all of the '#london2012' messages, such as '#london2012', '2012', and 'olympics', receive a lower score from the parsimonious model, and almost none of these

[1] https://dev.twitter.com/docs/api/1/get/search

Figure 2. Wordcloud of all #london2012 Twitter search results, showing the tweets containing the term 'torch'

Figure 3. Wordcloud of #london2012 Twitter search results containing the term torch, showing the tweets containing the term 'sheffield'

words occur in the 'torch' wordcloud. Also general words that occur frequently in all of the messages, such as 'get', and 'will' are filtered out. Instead the cloud contains words that occur more frequently in the subset of messages that contain the word 'torch', for example some of the cities that the torch passes through such as Sheffield, Leeds and Manchester. Every result in this cloud by definition contains the word 'torch', therefore it takes a prominent place in the wordcloud. You can choose to not show the word 'torch' in the wordcloud by specifying it as a stopword on the input screen.

Clicking on a term in the wordcloud has the same effect as query expansion, i.e. adding that term to your query and retrieve another set of results. When you use the Twitter API to search Twitter without query operators, only results will be returned that contain all of the search terms in the Tweet, username or hyperlink. This means adding a term to your query will not lead to more search results. Only if you remove the original query terms, other results will be returned.

Observations

We have not had the chance to evaluate our application through means of a user study. However, we do want to point out the following observations. Given the nature of our data, i.e. a collection of tweets, there might be some improvements possible that exploit this particular type of data. Tweets can contain special elements in the text, namely usernames, hashtags, links, and emoticons. We make the following observations:

- Usernames and hashtags are currently considered in the sense that we remove all punctuation except the characters '@' and '#' which are the indicators of usernames and hashtags respectively. There is an option to generate wordclouds containing only usernames, or only hashtags. In the default settings usernames and hashtags are included as is in the wordcloud. For future work we want to discuss and investigate two open issues:
    1. Can a word with a hashtag be considered as the same word without the hashtag? While a hashtag term does not always have to be a real word, e.g. #london2012, in many cases it is, e.g. #london. For the wordcloud

should the terms ‘london’ and ‘#london’ be merged? Sometimes usernames are used in a similar way as hashtags to address companies, e.g. in this tweet: ‘Ambush marketing at the Olympics! Well played, @Nike. bit.ly/N4zAUc #London2012’.

2. A related issue is the importance or term weights of usernames and hashtags. Is a hashtag a stronger signal, and should it therefore be featured more prominently in the wordcloud? Similarly for usernames, but usernames could also be considered a weaker signal, so should they be featured less prominently?

Both of these questions can also be considered when you want to optimize a retrieval algorithm.

- Besides the ‘@’, and ‘#’ all other punctuation is removed during text preprocessing. This means all emoticons like ‘:)’ are removed. Sometimes these emoticons are used as indicators of sentiment, i.e. tweets containing ‘:)’ are classified as positive messages, and tweets containing ‘:(’ as negative messages. In this sense the emoticons do indeed represent valuable information that could be included in the wordcloud. When an emoticon appears in the wordcloud, clicking on it can give you all the messages associated with for example a positive emoticon.

Feedback from users is required to determine the most useful improvements for the application.

4 Conclusions

In this paper we have shown how wordclouds can be used to summarize and navigate search results, and in particular Twitter search results. Wordclouds are a quick way to summarize and get a first overview of large amounts of data. Using human observation skills it is easy to zoom in on a group of messages in which you are interested, i.e. all messages that contain a specific term from the wordcloud. In future work we would like to evaluate the usefulness of wordclouds for navigation and summarization of search results in a user study.

5 References

[1] T. Gottron. Document Word Clouds: Visualising Web Documents as Tag Clouds to Aid Users in Relevance Decisions. In M. Agosti, J. L. Borbinha, S. Kapidakis, C. Papatheodorou, and G. Tsakonas, editors, *ECDL*, volume 5714 of *Lecture Notes in Computer Science*, pages 94–105. Springer, 2009.

[2] D. Helic, C. Trattner, M. Strohmaier, and K. Andrews. Are tag clouds useful for navigation? A network-theoretic analysis. *IJSCCPS*, 1(1):33–55, 2011.

[3] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious Language Models for Information Retrieval. In *Proceedings SIGIR'04*, pages 178–185. ACM Press, New York NY, 2004.

[4] R. Kaptein, D. Hiemstra, and J. Kamps. How Different are Language Models and Word Clouds? In *Advances in Information Retrieval: 32nd European Conference on IR Research (ECIR 2010)*, volume 5993 of *LNCS*, pages 556–568. Springer, 2010.

[5] B. Y.-L. Kuo, T. Hentrich, B. M. Good, and M. D. Wilkinson. Tag clouds for summarizing web search results. *Proceedings of the 16th international conference on World Wide Web WWW 07*, 196:1203, 2007.