Hashtag Similarity

Our attempt to design a metric for hashtag similarity focuses primarily on the semantic of a hashtags. A hashtag is deeply connected to a collection of related thoughts and ideas; they serve as links within a broad social discourse, connecting smaller spheres of topics and trends together. To ignore the semantics when comparing the similarity of hashtags would be to ignore the significance of their purpose.

In order to use a semantic based metric we must look beyond the 'physical' characteristics of a hashtag, such as the length, the characters it contains or even the popularity of the tag. While these properties can be useful, they do not reveal much in terms of a hashtag's meaning. We can gain insights into the semantic of a hashtag by looking at the context in which a certain hashtag occurs, namely the tweets it belongs to. By inspecting the collection of tweets relating to a single hashtag, we are able to infer to a deeper meaning of the hashtag.

The problem of measuring similarity of two hashtags now becomes the problem of measuring the similarity of the collections of tweets relating to each tag. The method of calculating document similarity is used frequently in search engines, and it is the approach we have employed in our project. A *document* is a collection of *terms*, where a term is simply a word. The underlying idea is to compare the term frequencies of two documents; if document A and document B contain similar frequencies of like terms, then they are considered to be similar documents. Given the term frequencies of a document, we are able to construct a vector for this document and view it within a vector space. Given two vectors (documents), we can calculate their similarity by measuring the angles between them. Very similar documents will be represented by two vectors pointing in the same direction, very dissimilar documents will be pointing in opposite directions. Each unique term corresponds to a single dimension in this vector space, thus the dimension of the vector space is the number of unique terms in the collection of all documents.

In regards to a hashtag, a document correlates to the collection of all tweets containing this hashtag, thus we construct a vector for each hashtag. The angle between two vectors therefore describe the similarity between two hashtags. It is important to note that some preprocessing of the data is required before constructing the vectors. A description of the whole process follows:

1. We find all tweets containing at least one hashtag and add each unique term to a set. This allows us to define a minimal vector space, that is, a vector space with the minimum amount of dimensions.
2. For each hashtag, we find every tweet containing this hashtag, and concatenate them together to form a document.
3. We remove certain terms that are irrelevant, such as hashtags, links and twitter handles.
4. For each hashtag document, we determine the term frequencies. This forms the vector of the hashtag. Note that all vectors must have the same dimension (the dimension of the vector space in step 1); if a term doesn't exist in the document, the value for that dimension is 0.
5. Next we normalise the vector by dividing each column of the vector by the total number of terms in the related document. A normalised vector has the property that sum the columns of the vector is equal to 1.
6. We then scale each value of the vector by the *inverse document frequency:* Since some terms occur more frequently than others (eg. 'the' compared to 'happiness'), these terms can severely outweigh the less frequent terms. More frequent terms are therefore weighted more lightly, and less frequent terms are weighted more heavily.
7. Now we have our processed vectors. We can measure the angles between two vectors to determine their similarities.

In order to integrate the vectors into the k-means algorithm, we simply use the angle calculations as our 'distance' function. Initially, k vectors are chosen to be the centroids of the clusters. Then

each vector is assigned to the centroid with the most similar angle. In the next iteration the new centroids for each cluster needs to be determined: this is the average of the cluster. We calculate this by finding the average vector of the given cluster. Then, when we redistribute the vectors to the clusters again, we compare each vector to the new centroid.

Although the our k-means program functions correctly, it is not immediately clear if a given cluster contains the most similar hashtags. A potential reason for this is that the vector space is too sparse; that is, there is not enough data to construct meaningful vectors for each hashtag. In our research on this method, meaningful results were obtained with sample spaces containing hundreds of thousands of tweets, where our database only contains a few thousand tweets. Furthermore, the majority of hashtags in our dataset only occur once, and therefore their vectors were constructed with an extremely small amount of data. We believe that the best results are achieved, when each hashtag has a relatively large collection of tweets related to it. Understandably, we knew it would be an ambitious task to use a metric based on semantics, and although it is unclear if are efforts were successful, we are satisfied with our attempt.