Hashtag Processing for Enhanced Clustering of Tweets

Dagmar Gromann
Artificial Intelligence
Research Institute (IIIA-CSIC)
Campus de la UAB,
E-08193 Bellaterra, Spain
dgromann@iiia.csic.es

Thierry Declerck
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken, Germany
declerck@dfki.de

Abstract

Rich data provided by tweets have been analyzed, clustered, and explored in a variety of studies. Typically those studies focus on named entity recognition, entity linking, and entity disambiguation or clustering. Tweets and hashtags are generally analyzed on sentential or word level but not on a compositional level of concatenated words. We propose an approach for a closer analysis of compounds in hashtags, and in the long run also of other types of text sequences in tweets, in order to enhance the clustering of such text documents. Hashtags have been used before as primary topic indicators to cluster tweets, however, their segmentation and its effect on clustering results have not been investigated to the best of our knowledge. Our results with a standard dataset from the Text REtrieval Conference (TREC) show that segmented and harmonized hashtags positively impact effective clustering.

1 Introduction

Social media and microblogging platforms continuously produce a wealth of information. The microblogs on Twitter, i.e., tweets, have been mined for nearly everything ranging from the detection of adverse drug reactions (O'Connor et al., 2014) to emergency response (Toriumi and Baba, 2016). One interesting problem in tweet mining is the automated detection of the tweet's topic. Hashtags have been found to be approximate indicators of a tweet's topic(s) (Rosa et al., 2011; Bansal et al., 2015; Zubiaga et al., 2011; Declerck and Lendvai, 2015) as they serve the purpose to point to a previously specified or emerging content. Based on the character limit of the platform, they are heavily used, which is, however, also the reason why they are frequently composed to save space. Different hashtags are combined to create a new topic reference, such as "#California#Drought", or words within hashtags are concatenated to reference a specific event, e.g. "#PoliticsandCurrentEventsCarolineKennedyMichelleObama".

Complex hashtags are marked by heavy concatenation (see example above) and terminological variation (e.g. "#IranDeal" and "#IranNuclearDeal"). Our research is based on the assumption that preprocessing those concatenations can improve topic identification results of tweets. To evaluate this assumption, this paper presents a method to segment concatenated hashtags and then uses them in a spectral clustering process to group tweets by their topic. We compare the results thereof with clustering results without preprocessing hashtags. We investigate if such a hashtag processing can improve categorizing tweets by topic by comparing the results of both clustering processes. Analysing the internal semantic structure of hashtags holds the promise to predict hashtags for tweets that do not use any hashtags based on the terms in the tweet and a given inventory of potential hashtags used within the same time period. It also facilitates the identification of terminological variation in tweets, which can be useful in scenarios relying on terms, such as emergency response or event detection.

2 Related Work

Hashtags have been previously used as approximate topic indicators in tweets (Rosa et al., 2011; Kapanipathi et al., 2014; Bansal et al., 2015). Rosa et al. (2011) apply supervised and unsupervised clustering algorithms to group tweets by topic based on a gold standard created from assigning a set of hashtags to specific topics. It has been shown before that clustering algorithms using hashtags as features produce good results (Rosa et al., 2011). However, hashtags have been

Proceedings of Recent Advances in Natural Language Processing, pages 277–283,
Varna, Bulgaria, Sep 4–6 2017.

treated as coherent units and their internal structure has not been considered in the clustering process. In this paper we first segment concatenated hashtags to facilitate the detection of similarity between hashtags, e.g. "#IranDeal" and "#IranNuclearDeal".

Hashtags also contain named entities. Their segmentation and linking to entities in knowledge bases has been studied before (Bansal et al., 2015; Kapanipathi et al., 2014). Kapanipathi et al. (2014) semantically enrich tweets with knowledge base content in order to represent user interests hierarchically. This is achieved by classifying named entities extracted from tweets into Wikipedia categories and representing them as a Hierarchical Interest Graph. In contrast to Kapanipathi et al. (2014) and Bansal et al. (2015), we do not limit our approach to entities and instead are interested in any type of term, within or outside of hashtags, while concentrating on the latter in this study with a focus on concatenated hashtags.

3 Dataset

To compare the clustering of tweets with preprocessed and non-preprocessed hashtags, we required a dataset with high-quality topic classifications for each tweet as a ground truth label for our method. Thus, we opted for a gold standard resource where each tweet is thematically classified. A Text REtrieval Conference (TREC-2015) shared task on identifying interest profiles of micro-blogging users (Lin et al., 2015) provides a set of twitter IDs, user-names, and dates for participants to stream the tweets in real-time during the task. When we streamed the tweets based on that list at a later moment in time, not all of them were available. In total, 6,187 tweets from the original dataset were annotated by 6 raters with one of 51 topics ranging from general topics, such as "self-driving cars" or "polar icecap melting", to more time-specific events, such as "Special Olympics 2015" or "Iran nuclear agreement" that was underway and reached in 2015. From those annotated tweets, 5,141 were still available when we streamed the data. Since our main interest in this publication is on hashtags, we limit the dataset to tweets containing hashtags which results in 2,053 tweets.

4 Method

To classify tweets by their topic, the proposed method relies on spectral clustering. Based on the assumption that the segmentation of concatenated hashtags improves this tweet classification, we compare clustering results with and without preprocessing (complex) hashtags. In the long run, we aim at a method for assigning topic labels to the resulting clusters by mining their contents for all types of concatenated sequences and aligning them to knowledge bases.

4.1 Data Preprocessing

To reduce the noise and allow for a separate handling of hashtags, a number of preprocessing steps are performed as described in this section.

URL handling: Instead of URL removal, they are replaced by identifiers and stored in a URL repository, e.g. "https://t.co/UrIygTXM0O" is replaced by "url30075" in all its occurrences. Thus, semantically rich URLs can be utilized as features in the clustering process. Furthermore, other punctuation can be processed without changing any punctuation in URLs.

Marked phrases: Similar to the URLs, hashtags (marked with '#') and replies (marked with '@') have a distinct semantic role and meaning in a tweet and are stored in a separate repository without their marking signs '#' and '@'. We remove those signs since in the segmentation process they would only be attached to the first word.

Stop word removal: The overall frequency of certain stop words, such as articles, is relatively high, while their level of informativeness is very low. For this reason we remove stop words from the microbologging contents in our dataset. Punctuation other than '#', '@', '_', and '-' are considered stop words and are removed from the corpus following Rosa et al. (2011) and for the same reason that stop words are removed. For the purpose of this study we also remove emoticons, which includes hashtags containing only emoticons.

4.2 Hashtag Harmonization

Concatenated hashtags represent an issue for similarity measures since a higher similarity is presumed if a phrase is represented without whitespace than with (Antenucci et al., 2011). But, very often, concatenated hashtags do not follow

a consistent way of being built. In our corpus, we have for example "#CaliforniaDrought", "#CADrought", "#cadrought" and more variants of the same concept expressed by using a hashtag. There are more complex examples of this type of term variants, such as "#IranTalksVienna", "#IranTalks", "#IranNuclearDeal", "#Irandeal", "#dealwithiran", "#DisasterIranDeal", "#NoNuclearIran", "#NoIranDeal", "#StopIranDeal", "#stopirandeal", "#StopIranRally", "#badirandeal", etc. Frequently, hashtags are not just attached to the end of a tweet but incorporated into its sentential structure, such as "#IranDeal lifts sanctions". All of those variants relate to one topic, namely the Iran nuclear agreement between Iran and a group of world powers, but some also express opinions about this topic. There is thus a need to segment and normalize hashtags in order to improve measures of similarity, as has also been suggested by other approaches to using Twitter data (Declerck and Lendvai, 2016; Rosa et al., 2011; Kapanipathi et al., 2014).

4.2.1 Orthography

The simplest harmonization step consists in adjusting different typographical versions of a hashtag. A straightforward approach is the lowercasing of the text included in a simple (i.e. not concatenated) hashtag. So for example, in our corpus, the hashtag "#Iran" occurs 236 times, the hashtag "#iran" 28 times and the hashtag "#IRAN" 3 times. After this transformation step, the harmonized hashtag "#iran" will thus occur 267 times. As a result from this simple harmonization process, the "#iran" hashtag can be consistently used as a potential semantic label for (a group of) tweets.

We include in the transformation process the elimination of certain punctuation signs that are attached to the hashtag, like for example "#Iran," or "#Iran:". The lowercasing step is limited in a first phase to simple hashtags, since we need the hashtags using the CamelCase notation as an initial data for the segmentation step. Lowercasing can subsequently be applied to the results of the segmentation process.

4.2.2 Segmentation

In the examples shown in the introduction to this section, we can observe the possibly very large number of concatenated hashtag variants expressing one topic. We aim at providing a segmentation and harmonization of the components of such hashtags in order to reduce the number of their variants. We apply for this a rather conservative approach in order to avoid the segmentation of hashtags like "#KnockKnockLive" into three "new" hashtags "#Knock", "#Knock" and "#Live" ("Knock Knock Live" is the name of a television series, and should therefore not be segmented into three hashtags.).

A first candidate for segmentation are hashtags that are expressed using a CamelCase notation, such as "#IranDeal" (or even "#KnockKnockLive", which at the end should not be segmented). It is straightforward to segment such hashtags, the segments being defined as the sequences starting with a capital letter. As a preliminary filter for avoiding unwanted segmentations, we request that at least one of the resulting segments occurs as a standalone hashtag (for example "#Iran" in the case of "#IranDeal" or "#NoNuclearIran") in the corpus, and that all the resulting segments are listed in the Unix dictionary `words` file[1]. This way, the segmentation of "#KnockKnockLive" in three components is avoided, as none of the potential segments occurs as a standalone hashtag in the corpus. The algorithm will have to be refined for dealing with other and larger corpora.

This approach offers a basis for the correct segmentation for a set of concatenated hashtags that contain only lowercase letters, for example "#irandeal", as those are typographical variants of the hashtags in CamelCase notation. To give an example on how a concatenated hashtag is segmented, here with the hashtag "#irandeal", which after processing is internally represented as a feature structure:

```
'irandeal':
   {'freq': 9, 0: 'iran', 1: 'deal'}
```

The reader can see that we encode also the order of the components of the segmented hashtag.

As a result of this segmentation process, we can augment the number of times the strings "Iran", "iran" or "IRAN" have been detected within an hashtag from 267 to 468, thus significantly increasing the evidence that the corpus has "iran" as a main topic[2].

[1] See for more detailshttps://en.wikipedia.org/wiki/Words_(Unix).

[2] For the sake of the description we have been keeping the # sign in many examples. But for the clustering step, described on 4.3, we remove this marker in order not to have

Current work is dedicated to improving the grouping of terms extracted from the segmented concatenated hashtags, so that for example "#dealwithiran" can be properly associated with "#IranNuclearDeal" and similar hashtags. This will be based on structuring 'sub-topics' associated with the 'main' topic, such as "iran". We expect from this step an additionally improved clustering performance. We will also study the possibilities to extend the coverage of "save" and meaningful hashtag segmentations.

4.3 Clustering

To make natural language data useful in an automated fashion they need to be grouped and categorized. The state-of-the-art method to address this task of semantic categorization is unsupervised clustering (Baroni et al., 2014). We perform clustering based on the normalized spectral clustering algorithm proposed by Ng et al. (2001) that has been effectively applied to various lexical acquisition and classification tasks (e.g. Xu and Ke, 2016; Shutova et al., 2016). Spectral clustering is particularly attractive since it is reasonably fast and treats data clustering as a graph partitioning problem. Calculating the distance based on a graph rather than a pair-wise comparison of distances between points is particularly well suited for smaller numbers of clusters, such as the 51 preset clusters in our case based on our dataset.

Even though supervised clustering methods tend to perform better (Shutova et al., 2016), we believe and want to show that simple string harmonization and term variant detection methods can improve both unsupervised and supervised methods. In this paper we start with the improvement of the results of unsupervised methods, while the evidence for supervised methods is yet to come.

The input data are strings consisting of words and urls as well as specially marked words, such as hashtags, replies, and retweets. Some normalization steps are performed to the overall clustering processes as described in Section 4.1. The terms to be clustered are the tweet IDs and their preprocessed contents represent the feature vector with their raw frequencies, e.g. tweet ID "626157993101512704" has the feature vector {when: 37, drought: 489, California: 350, ...} representing the tweets content words and their relative raw frequency. The raw frequency is then turned into a distance measure in the process of creating a similarity matrix.

Computing a similarity matrix depends on the choice of semantic distance measure that is best for the given data. The most commonly used ones are Term Frequency-Inverse Document Frequency (TF-IDF), Positive Pointwise Mutual Information (PPMI), Kullback-Leibler divergence, string edit distances such as the Jaccard distance, cosine distance, and as of late APSyn (Santus et al., 2016). The Kullback-Leibler divergence is a useful feature to measure mutual information, that is, the concordance among sub-units of phrases (Lin et al., 2015). Thus, it has less bias towards rare-occurring phrases and is particularly adequate for the comparison of compounds. A symmetric and smoothed version of the Kullback-Leibler is the Jensen-Shannon divergence (JSD). Since the similarity matrix represents a weighted but undirected graph, the JSD is more adequate than Kullback-Leibler and for the two feature vectors v_i and v_j is defined as:

$$JSD(v_i||v_j) = \frac{1}{2}D(v_i||M) + \frac{1}{2}D(v_j||M) \quad (1)$$

where D represents the Kullback-Leibler divergence and M is defined as the average of v_i and v_j. We adopt the successful creation of a similarity matrix by Shutova et al. (2016) and define the similarity w_{ij} as

$$w_{ij} = e^{-JSD(v_i,v_j)} \quad (2)$$

We tested with the well-known TF-IDF and PPMI as well as with JSD, but only explain JSD here in detail because it is less well-known than the other two. The similarity matrix build on the respective semantic distance measure and the predefined number of resulting clusters represent the input to the spectral clustering algorithm presented as Algorithm 1.

Based on the input matrix a degree matrix is formed as detailed in Algorithm 1. We followed von Luxburg (2007) and tested the ϵ-neighborhood, k-nearest neighbor (knn), and a fully connected graph building method on our dataset. The difference between the degree and the weighted adjacency matrix form the graph Laplacian L. The normalized matrix of eigenvectors of the normalized L is then used as input to the k-means algorithm. The algorithm provides the

it considered in the frequency count of the resulting feature vectors.

Algorithm 1 Spectral Clustering

1: **Input:** Similarity matrix $S \in \mathbb{R}^{nxn}$, number of k clusters
2: Construct a degree matrix D where $d_i = \sum_{j=1}^{n} w_{ij}$ and d_{ij} = 0 if i $\neq$ j
3: Construct a similarity graph and its weighted adjacency matrix W
4: Construct a graph Laplacian $L = D - W$
5: Compute the normalized Laplacian $L_{sym} := D^{-1/2}LD^{-1/2}$
6: Compute the first k eigenvectors v_1,...v_K of L_{sym} and write them as columns into the matrix $U \in \mathbb{R}^{nxk}$
7: Compute the matrix $T \in \mathbb{R}^{nxk}$ from U by normalizing that is set $t_{ij} = u_{ij}/(\sum_k u_{ik}^2)^{1/2}$
8: Let y_i be the vector corresponding to the i^{th} row of T
9: Cluster the points $(y_i)_{i=1,...,n}$ with the k-means algorithm into clusters C_1,...,C_k
10: **Output:** Clusters C_1,...,C_k with $C_i = \{j|y_j \in C_i\}$

number of clusters that was initially provided as input. To optimize this variable, we experimented with different sizes of k to be detailed in Section 5. The result of each cluster is the tweet IDs they contain as well as all the preprocessed words that each tweet ID represents. The preprocessing described in Section 4.1 is applied to all tweets and all clustering runs. The decomposition and harmonization of hashtags is only performed for the second clustering run as described below to compare the effect this processing has on the clustering.

5 Results

Before we detail the comparison of the cluster results, we provide some basic statistics on the harmonization of hashtags.

5.1 Hashtag Harmonization Results

In the corpus we had a total of 3.893 hashtags, out of which 2.697 contained at least one capital letter and 1.196 were exclusively in lowercase. In total the number of segmented hashtags in CamelCase notation amounts to 1.024 and to 91 for hashtags containing only lowercase notation. Lowercase refers to hashtags in which the concatenation is not clearly marked by capitalization, such as "#horseracing". In contrast, in case of CamelCase notation, the capitalization within a sequence of characters indicates potential word boundaries, such as "#HorseRacing". Both hashtags refer to the event "horse racing" but different techniques need to be applied for their segmentation, as described in Section 4.2.2.

At first sight the results for the segmentation of lowercase hashtags are accurate, while we need to improve the accuracy of the segmentation process for the hashtags in CamelCase notation. We expect then another improvement of the results presented in Section 5.2

5.2 Clustering Results

Given the ground truth labels of the TREC gold standard dataset and the prediction labels of our clustering algorithm, mutual information can be used to measure the agreement of their assignment. Adjusted Mutual Information (AMI) has been proposed (Vinh et al., 2009) as a measure from information theory that can be used to compare clustering overlaps. It is normalized against chance and particularly adequate for unbalanced reference clustering with varying cluster sizes (Romano et al., 2016).

In Table 1 an AMI-based comparison of the three graph building methods *knn*, ϵ, and *fully connected* with the three main similarity measures is illustrated. The best result is achieved by the combination of a *knn* graph with TF-IDF as similarity measure with 0.754 as highlighted in Table 1. The other graph building methods, however, perform equivalent to *knn* with TF-IDF. We represent the difference between clustering without and with preprocessed hashtags in the columns *unsegmented* (uns.) respectively *segmented* (seg.). Numbers in the segmented column always slightly exceed results without preprocessed hashtags, which shows that the accuracy of clustering can be improved by a basic and straightforward approach to normalizing and segmenting hashtags, even if only slightly.

	knn		ϵ		**fully conn.**	
	uns.	seg.	uns.	seg.	uns.	seg.
TFIDF	0.687	**0.754**	0.678	0.750	0.677	0.737
PPMI	0.664	0.707	0.676	0.701	0.658	0.701
JSD	0.486	0.531	0.436	0.468	0.442	0.487

Table 1: AMI Unsegmented and Segmented Clustering Results for Tweets

To explain our results in more detail, we provide two example sentences below with their re-

spective feature vector. Without segmentation, it would have been unlikely that "#drought" in example sentence 1 would have achieved a high similarity measure with the concatenated and camelcased hashtag "#CaliforniaDrought" in example sentence 2. With the preprocessing, their representation in the feature vector is more similar and both are grouped in the same thematic cluster.

```
Example sentence 1: ``When there is
drought in California, people just #paint
their #lawn https://t.co/I0zuBVAXhn
#lifehack #drought''
Feature vector 1: 626157993101512704:
['When', 'drought', 'California',
'people', 'paint', 'lawn', 'lifehack',
'drought', 'url34452']

Example sentence 2: ``In reality the
apocalypse has already happened, but it
came quietly & slowly, so we didn't
notice. #CaliforniaDrought
http://t.co/UOZbMp4yDA''
Feature vector 2: 623383930763366400:
['In', 'reality', 'apocalypse', 'already',
'happened', 'came', 'quietly', 'slowly',
"didn't", 'notice', 'california',
'drought', 'url4850']
```

We were also interested in whether a similar improvement could be achieved when clustering is exclusively performed based on hashtags. A comparison is presented in Table 2 where we can see that this assumption is true other than in the case of the ϵ graph building method. The comparison is only based on JSD since the other two similarity measures returned results lower than 0.2.

	knn		ϵ		**fully conn.**	
	uns.	seg.	uns.	seg.	uns.	seg.
JSD	0.270	0.332	0.371	0.357	0.404	**0.437**

Table 2: AMI Unsegmented and Segmented Clustering Results for Hashtags

6 Discussion

While the first results obtained are encouraging, we are aware that our approach to the harmonization of (complex) hashtags needs to be refined. We need for example to avoid the segmentation of #YouTube or #FootBall. While this has been already effectively implemented for the latter example, as the word "football" also occurs in the Unix dictionary `words` file, we will need to address the issue of dealing with named entities like "YouTube", since we cannot only rely on the fact that neither "#You" nor "#Tube" occur as standalone in the corpus under investigation. This is a place where we need to investigate the reuse of approaches dealing with entity linking in the field of Twitter data.

7 Conclusion and Future Work

The experiments conducted in this study on spectral clustering applied to tweets that have undergone basic steps to the segmentation of concatenated hashtags have supported our expectations that harmonized hashtags can lead to a better topic categorization of Twitter texts. This paper provides statistical evidence that the clustering results improve when using processed hashtags. Comparable improvements can be observed when clustering tweet IDs exclusively based on hashtags, leaving the rest of the tweet text aside, which stresses the central role played by hashtags for the categorization of topics of tweets.

In terms of future works, the current experiments need to be extended to more and larger corpora and also the segmentation algorithm could be refined. For example, we think of a hierarchical structure of topics and sub-topics resulting from the segmentation of compounds into main components and modifying components that can be identified by means of dependency analysis. We plan to deal also with other types of compounds in use in micro-blog texts, but concatenated hashtags have the advantage to explicitly mark the combination of words.

A next step will consist in enhancing our spectral clustering approach with graph-based knowledge for an effective semantic classification of tweets, annotating those automatically with DBpedia resources. By linking the resulting terms in the clusters to such a knowledge base, we can improve the identification of the meaning of the cluster and assign it a more accurate topic label.

Acknowledgments

This research has been partially funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 607062 /ESSENCE: Evolution of Shared Semantics in Computational Environments/ for the IIIA-CSIC contributions and for the DFKI contribution by the project /ALL-SIDES: Advanced Large-Scale Language Analysis for Social Intelligence Deliberation Support/ funded by the German Federal Ministry of Education and Research (BMBF).

References

Dolan Antenucci, GREGORY Handy, AKSHAY Modi, and Miller Tinkerhess. 2011. Classification of tweets via clustering of hashtags. *EECS* 545:1–11.

Piyush Bansal, Romil Bansal, and Vasudeva Varma. 2015. Towards deep semantic analysis of hashtags. In *European Conference on Information Retrieval*. Springer, pages 453–464.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*. pages 238–247.

Thierry Declerck and Piroska Lendvai. 2015. Processing and normalizing hashtags. In *RANLP*. pages 104–109.

Thierry Declerck and Piroska Lendvai. 2016. Towards the harmonization and segmentation of german hashtags. *Bochumer Linguistische Arbeitsberichte* page 10.

Pavan Kapanipathi, Prateek Jain, Chitra Venkataramani, and Amit Sheth. 2014. User interests identification on twitter using a hierarchical knowledge base. In *European Semantic Web Conference*. Springer, pages 99–113.

Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. 2015. Overview of the trec-2015 microblog track. Technical report, DTIC Document.

Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. 2001. On spectral clustering: Analysis and an algorithm. In *NIPS*. volume 14, pages 849–856.

Karen O'Connor, Pranoti Pimpalkhute, Azadeh Nikfarjam, Rachel Ginn, Karen L Smith, and Graciela Gonzalez. 2014. Pharmacovigilance on twitter? mining tweets for adverse drug reactions. In *AMIA annual symposium proceedings*. American Medical Informatics Association, volume 2014, page 924.

Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. 2016. Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research* 17(134):1–32.

Kevin Dela Rosa, Rushin Shah, Bo Lin, Anatole Gershman, and Robert Frederking. 2011. Topical clustering of tweets. *Proceedings of the ACM SIGIR: SWSM* .

Enrico Santus, Emmanuele Chersoni, Alessandro Lenci, Chu-Ren Huang, and Philippe Blache. 2016. Testing apsyn against vector cosine on similarity estimation. *arXiv preprint arXiv:1608.07738* .

Ekaterina Shutova, Lin Sun, Elkin Darío Gutiérrez, Patricia Lichtenstein, and Srini Narayanan. 2016. Multilingual metaphor processing: Experiments with semi-supervised and unsupervised learning. *Computational Linguistics* .

Fujio Toriumi and Seigo Baba. 2016. Real-time tweet classification in disaster situation. In *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, pages 117–118.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, pages 1073–1080.

Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17(4):395–416.

Zhiqiang Xu and Yiping Ke. 2016. Effective and efficient spectral clustering on text and link data. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, pages 357–366.

Arkaitz Zubiaga, Damiano Spina, Víctor Fresno, and Raquel Martínez. 2011. Classifying trending topics: a typology of conversation triggers on twitter. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, pages 2461–2464.