Information Sciences 505 (2019) 65–83

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Identifying topic relevant hashtags in Twitter streams

Filipe Figueiredo*, Alipio Jorge

FCUP, Universidade do Porto / LIAAD - INESC TEC, Rua Campo Alegre, Porto 1055/4169-007, Portugal

ARTICLE INFO

Article history:
Received 12 December 2018
Revised 8 July 2019
Accepted 14 July 2019
Available online 15 July 2019

Keywords:
Text mining
Topic modeling
Latent Dirichlet Allocation
Support vector machines
Twitter
Hashtag recommendation

ABSTRACT

Hashtags have become a crucial social media tool. The categorization of posts in a simple and informal way helps to spread the content through the web. At the same time, it enables users to easily find messages within a specific topic. However, the flexibility provided to use and create a hashtag carries some problems. Equivalent expressions, like synonyms, are handled like entirely different words. On the other hand, the same hashtag may refer to different topics. In this paper, we present TORHID (Topic Relevant Hashtag Identification), a method that employs topic modeling with the purpose of retrieving and identifying hashtags relevant to a specific topic in Twitter streams, starting from a *seed* hashtag and resorting to a classifier to remove non relevant hashtags. The result is a network of hashtags related to the *seed*, that we can use to deepen the initial search.

1. Introduction

Twitter has become one of the most popular microblogging services in the world [6] with over 300 million active users every month [7]. The social network allows its users to communicate by posting short messages called âætweetsâg, which consist in pieces of text up to 280 characters long (originally only 140 characters). Due to the massive amount of users, around 8000 tweets are posted every second [23], which, despite the relatively small size of the messages, results in an astounding amount of information spread through the internet every day.

The original motivation for our work was to retrieve data from posts on social media in order to support a project (RECAP[1]) whose objective was to study physical conditions, on any stage of life, that could be related to a premature birth. Obstacles in determining the most advantageous hashtags for querying Twitter to collect this information lead to the development of a tool for hashtag recommendation. After improvements and adaptations to perform appropriately for other topics discussed in the platform, we denominated this system as TORHID (Topic Relevant Hashtag Identification).

Most of the functionality and tools of Twitter that make it so important in public communication today were user-led innovations, later integrated into the service itself. One of these distinctive components is the hashtag. Because of Twitter's popularity, it became burdensome to search for relevant information about a specific topic on the service. For that reason, users started applying dynamic, user-generated tagging signaled by a "#" to their tweets as a way to make them publicly available, categorized and, consequently, easier to search by other users interested in that topic.

While the current hashtag implementation brings several advantages for its users, because of its easiness of use, there are still some issues related to it when searching for content of a specific topic. Since anyone can use any word or expression as a hashtag limited by the poster's perspective and imagination and by Twitter's imposed 280 characters cap, it may be

* Corresponding author.
E-mail addresses: filipe.m.figueiredo@inesctec.pt (F. Figueiredo), alipio.jorge@inesctec.pt (A. Jorge).

[1] https://recap-preterm.eu/.

https://doi.org/10.1016/j.ins.2019.07.062
0020-0255/

difficult for a user to find every tweet relevant to his search. Hashtags are often ambiguous and some concepts they express may be sub-topics of others or better represented by a different hashtag. On the other hand, hashtags are dynamic in a sense that their popularity and meaning is constantly fluctuating. Since only one hashtag can be searched at a time, it may not be immediately clear which terms are the best to search for a certain topic.

In this paper, we propose techniques to identify and retrieve relevant hashtags for a given topic. Firstly, we generate a graph of hashtags collected from searching one *seed* hashtag and the retrieved hashtags recursively. Since an unrestrained expansion of the network inevitably reaches off-topic descendent nodes, we employ topic modeling to automatically disambiguate tweets based on their topical context and a support vector machine to filter non relevant tweets and corresponding hashtags. This method achieves an assortment of hashtags relevant to the original topic and the relationships among them in a network represented through a weighted directed graph. These hashtags may later be used to improve future searches on Twitter about the subject.

Fig. 1 depicts an example of the enhancements achieved by applying the proposed pruning process to a network of hashtags: the structure on top represents a network of hashtags collected without restrains for the seed *#Preterm*, where a considerable portion of the nodes is off-topic; the structure on the bottom represents the final pruned network, with relevant hashtags for the same seed.

The remainder of this paper is structured as follows: Section 2 describes the related work. Section 3 describes the methodology used to collect the tweets, identify the relevant hashtags for a determined topic and prune the graph. Section 4 describes the evaluation of the results and a comparison with a different approach. The conclusions are presented in Section 5.

2. Related work

The recent development and growth of social networks attracted a lot of attention from the research community. Twitter, as the most popular microblog service in the world, has become appealing due to the public availability of millions of short texts shared every day.

Previous research in Twitter hashtags mainly focused in recommending relevant hashtags to a specific tweet and not to a hashtag. Unlike the first keyword extraction systems [13,26], Mazzia and Juett [14] propose a method to recommend relevant hashtags, not limited to words already present in a tweet. The authors suggest considering every hashtag as a category and, correspondingly, the tagged tweets as labeled data, thus allowing the application of a naive Bayes model to calculate the maximum *a posteriori* probability of each hashtag class.

The approach proposed by Zangerle et al. [27] is to calculate the resemblance of tweets and attribute them a score based on a Term Frequency-Inverse Document Frequency (TF-IDF) scheme. Then, the hashtags are extracted and restricted to a final set of those with a score above a threshold. They experiment ranking this set based on the overall popularity of the tweets, on the popularity within the most similar tweets and on the resemblance with other tweets.

An improved version of the recommendation system proposed by Zangerle et al. was later suggested by Kywe et al. [10] which recommends hashtags also based on the user similarity. Given a user and a tweet, this methodology employs the TF-IDF scheme to select the most similar users, as well as the most similar tweets. The set of hashtags is then extracted from the most similar tweets and most similar users and ranked based on those metrics.

Efron [4] proposes a method that estimates the probability of every word in a previously collected data set to co-occur with every hashtag in a tweet and smooths the models through Bayesian updating with Dirichlet priors. The amount of information contained in every hashtag is later assessed through an Inverse Document Frequency (IDF) scheme and KL-divergence is used to compare the models of every hashtag with the one from the original query and rank them accordingly.

Latent Dirichlet Allocation (LDA) also become a popular tool for researchers attempting to process and organize hashtags. Zhao et al. [28] proposed the unsupervised topic model Twitter-LDA, a variation of LDA designed taking into consideration the length limitations of tweets. The method was used to discover topics from a sample of the entire Twitter in order to compare them with the ones found in New York Times. However, Hashtag recommendation was not one of the goals of the paper.

Godin et al. [8] apply Latent Dirichlet Allocation with the purpose of clustering previously collected tweets in a set of general topics. Given a new tweet, its underlying topic distribution is also generated and top keywords from the dominant subjects are recommended as hashtags. She and Chen [20] develop a supervised topic model-based solution for hahstag recommendation on Twitter, abbreviated TOMOHA. They employ an adaptation of Twitter-LDA [28] which considers hashtags as the labels of the local topics to generate a model capable of analyzing relationships among words, hashtags and topics of different posts.

Ramage et al. [19] apply a partially supervised learning model based on Labeled LDA to map sets of tweets in four dimensions (substance, status, social and style), with the purpose of profiling Twitter users and their habits for better follower recommendations. Mehrotra et al. [15] proposed a method to enhance LDA, employing different pooling schemes to aggregate tweets with the purpose of improving topics learned from Twitter content. Despite the scheme with better results relying in hashtag labeling, hashtag recommendation was not explored at this time.

Antenucci et al. [1] present methods to cluster hashtags in topic groups using a combination of co-occurrence frequency, graph clustering and textual similarity. These topic groups are used to classify a given tweet, based on the words it contains.

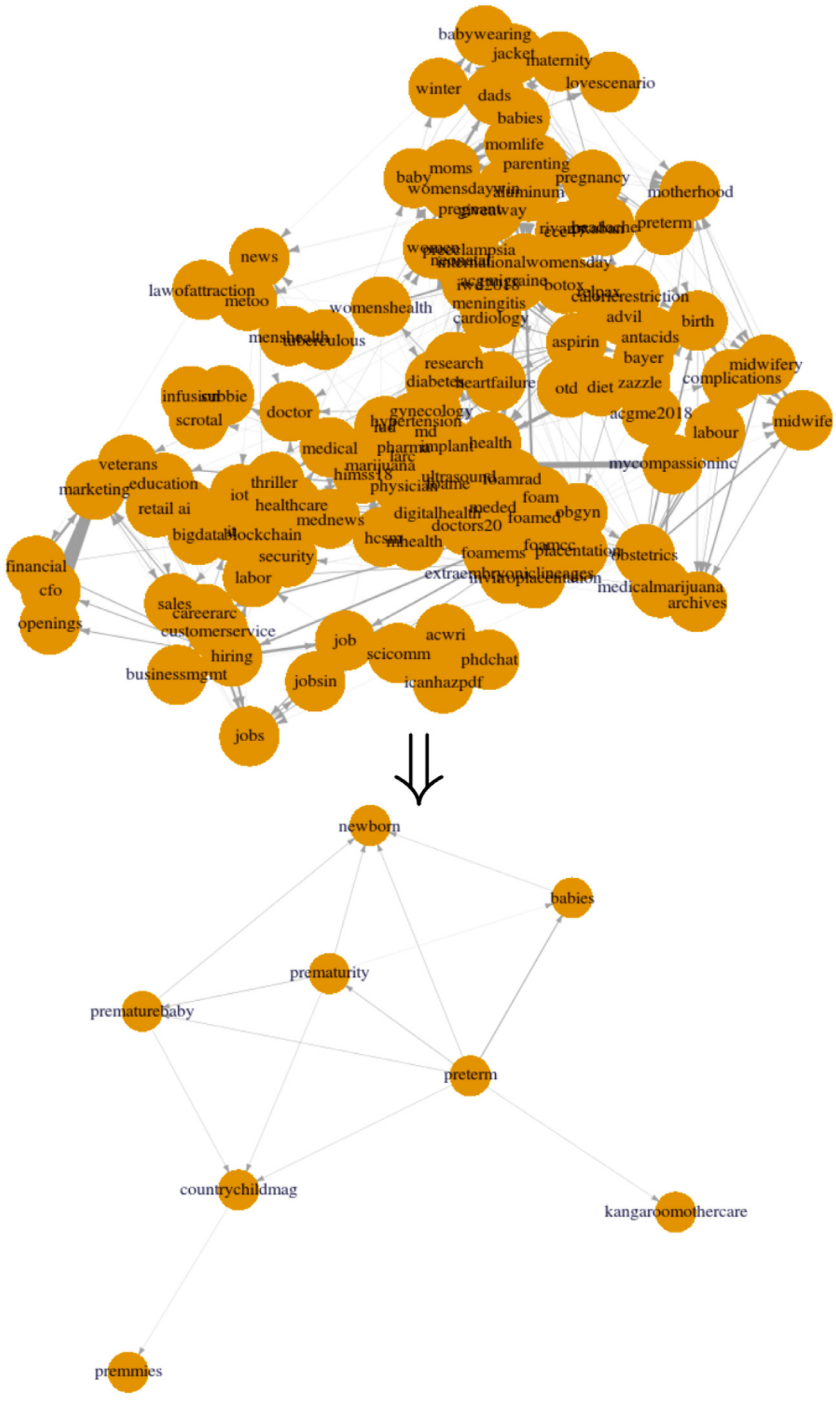

Fig. 1. Impact of the pruning process on a networks of hashtags related to *#Preterm*.

Attempting to improve user interactions with articles and publications, Xiao et al. [25] propose a method to recommend Twitter hashtags related to a keyword that represents a news related topic. They begin by collecting news articles from major agencies and employ an original Probabilistic Inside-Outside Log (P-IOLog) method to cluster them into vectors representing different topics. Tweets related to news which were published concurrently to the articles are also collected from a set of manually selected accounts and concatenated according to their hashtags, with the purpose of building a similar hashtag vector. The similarity between each news-topic vector and each hashtag vector is subsequently calculated.

Shi et al. [21] also propose a news related hashtag recommendation solution denominated *Hashtagger*, with the purpose of finding relevant hashtags for stimulating the dissemination of news articles. They adapt standard Information Retrieval (IR) methods to accept a news article as the initial query and a set of tweets representing their embraced hashtags as the documents. For each new article, a ranking is performed based on automatic query formulation, to retrieve candidate hashtags, and a previously trained pointwise learning-to-rank (L2R) method is subsequently used to score the relevance of each hashtag for each article. The same authors later propose *Hashtagger+* [22], an improved version of *Hashtagger* that employs cold-start algorithms to accelerate the recommendation process.

Otsuka et al. [18] suggest a method employing an adapted ranking system to recommend more current trending hashtags to new Twitter posts. Inspired by classic information retrieval methods, their process requires an earlier compilation of millions of tweets into two nested maps data structures: a Term to Hashtag Frequency Map (THFM), that maps each word with the frequency they co-occur with each hashtag; and a Hashtag Frequency Map (HFM), an analogous data structure that maps each hashtag with the respective term frequencies. Next, these mappings are manipulated by a Hashtag Frequency-Inverse Hashtag Ubiquity scheme (HF-IHU) to rank the hashtags in order of relevance to the tweet.

Different to the previous methods, Li et al. [12] focus on the search of content in Twitter by proposing hashtags for a keyword instead of a full tweet. The experiment constructs word embeddings [17] from a database with billions of words extracted from tweets. When querying for a keyword, every hashtag extracted from the tweets is ranked according to the cosine similarity score between their corresponding word-embedding vector and the embedding vector of the keyword itself.

As hashtags have become more and more important for online marketers and communicators, some websites like *Hashtagify*[2] have specialized in finding relevant hashtags to amplify their clients' reach and track their competitors. However, since these websites have a commercial nature, there is little or no information on their methodology of work from a scientific perspective.

Most of the previous approaches rely on static databases with millions of random tweets to train their classifiers. On the other hand, none of them takes into consideration the evolution of the recommended hashtags whose meaning and popularity change over time. For example, in 2010 the hashtag *#Johannesburg* would be considered relevant to the topic *#WorldCup*, considering that the FIFA World Cup was hosted in South Africa in that year; however, that would probably not be case in 2014 when the same event was held in Brazil. Therefore, in this paper, we propose a method to identify up to date hashtags relevant to a given topic, resorting to smaller data sets to train the classifier, which are easier to keep up to date.

3. Proposed method

Our goal is to develop a method that is able to collect relevant hashtags associated with a given topic, assuming that topic to be denoted by a hashtag itself. For the remaining of this paper, this hashtag representing the main topic will be referred to as the *seed*.

We based TORHID on two main ideas that differentiate it from current approaches: first, instead of employing a random sample of tweets, we will be using the Twitter Search API to collect posts containing a query, thus avoiding the acquisition of unnecessary data; in addition, instead of simply considering the tweets gathered from searching the *seed*, we will be extending our search with the hashtags found to be relevant to the topic. In this manner, we expect to also retrieve hashtags used alternately with the *seed* but rarely together in the same tweet, like variations of event names according to the current year.

Every time we search for hashtags in Twitter, the recovered tweets may contain hashtags with four types of relations to it: they may be unrelated; they may be synonyms or on the same topic; they may be a subtopic; or they may be a supertopic. While collecting subtopics generally doesn't entail any issues, when the algorithm recovers a supertopic it may result in the consequent return of its subtopics. If the supertopic is vague enough, its subtopics will probably be unrelated to the *seed*.

For that reason, it becomes necessary to find a way to separate the relevant hashtags we want, from the extra ones we collect unintentionally. Since correctly evaluating the relevance of a hahstag to another may be a challenging task even for a human being, we decided instead to evaluate the tweets containing it, thus providing some contextual reference. Our proposed approach is divided in two stages: firstly we collect enough tweets to train a classifier and then we collect a final sample of tweets and hashtags approved by that classifier. The relations between those hashtags can then be represented by a directed graph. The flowchart in Fig. 2 depicts a simplified overview of the proposed hashtag recommendation process.

3.1. Data collection and preprocessing

The first stage of our methodology is data collection, that is, obtaining an untreated assortment of related tweets. For this task, we used the R package `twitteR` [5], which provides an interface in R to the Twitter public available search API. Despite its constraints compared to other Twitter public API's, the use of `twitteR` allows us to search and collect tweets which are already somewhat related to the *seed*. This tool also allows us to restrict the search in time, location and language.

[2] http://hashtagify.me/explorer/about.

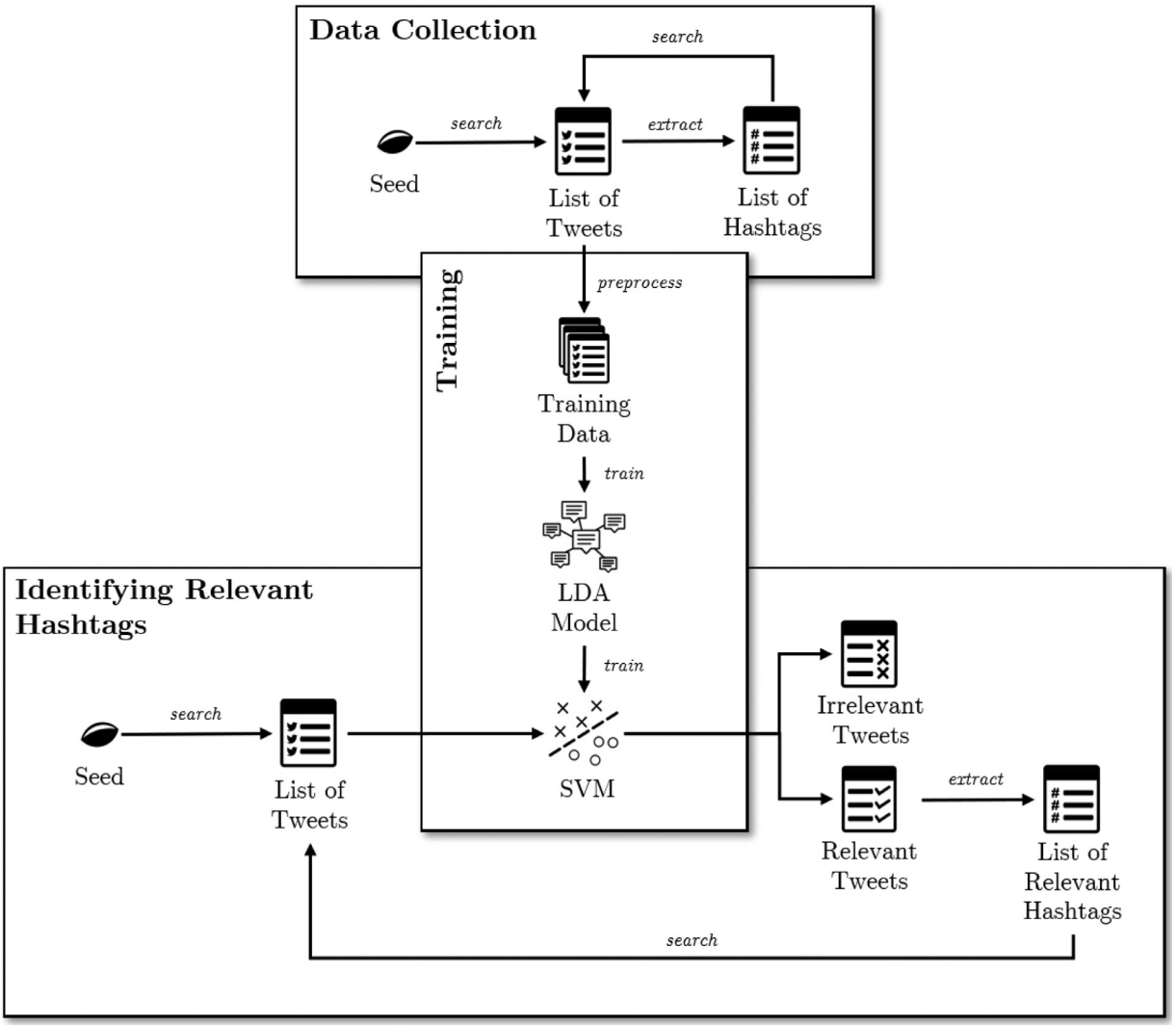

Fig. 2. Process of Hashtag recommendation.

Initially, we select the *seed*, which is a good representative of the topic that we want to explore. While the first *seed* for the topic is usually a conjecture, it is possible to replace it by a more adequate *relevant hashtag* returned by the algorithm. We proceed to search Twitter for the latest most popular tweets with that hashtag (internally determined by Twitter). We store this selection of tweets and scan them for new hashtags. The hashtags which appear a number of times above a threshold are kept in a queue. We then repeat the process but search for the hashtags in the queue, one at a time, instead of the *seed*, until we empty the queue or reach a limit on the number of searches (in order to prevent loops). For preventing the algorithm from searching the same hashtag twice, a list of previously searched terms is maintained and checked before adding new hashtags to the queue. The pseudo-code representation of the method described above to collect new tweets is shown in Algorithm 1 .

We perform several preprocessing steps to prepare the data for the topic modeling stage: all non-ASCII characters are removed; the text is converted to `utf-8`; urls are removed; all non-alphabetical characters are removed; all letters are converted to lower case; stopwords are removed; and word stemming is applied.

3.2. Training

If we were to scan the tweets collected by Algorithm 1, without any further restraints, we would be able to retrieve a large assortment of hashtags. However, only a small minority of those would actually be relevant to the intended topic. Since the expansion of the hashtag queue occurs without limitations, the network of hashtags will always reach nodes whose

Algorithm 1: Collecting Tweets.

Input: *seed*, *max.searches*, *threshold*
HashtagQueue ← initiate queue with *seed*
AllTweets ← ∅
while *HashtagQueue not empty* ***and*** *max.searches* > 0 **do**
  CurrentHT ← dequeue *HashtagQueue*
  NewTweets ← set of retrieved tweets containing *CurrentHT*
  AllTweets ← *AllTweets* ∪ *NewTweets*
  NewHashtags ← hashtags appearing in *NewTweets* more than *threshold* times
  HashtagQueue ← *HashtagQueue* ∪ *NewHashtags*
  max.searches ← max.*searches* − 1
end
store *AllTweets*

descendents' hashtags cease to have a sufficient degree of relation to the original topic, undermining the final set. For that reason, it becomes necessary to find a procedure to separate the relevant hashtags from the ones we collect unintentionally.

With the purpose of providing some context to the classifier, instead of directly evaluating the relevance of the hashtags to the topic, we will be classifying the tweets containing them. This way, even ambiguous words, concatenations of words and acronyms should have a better chance of being well classified. Furthermore, we decided to perform the classification process in place, to avoid the collection of unnecessary data and the consequent waste of resources.

We combine two well known techniques to classify tweets: LDA and SVMs. Latent Dirichlet Allocation (LDA) [2] is a generative model which assumes that underlying the data collection, exists a topic model with a given number of topics, which don't necessarily correspond to real life themes. Each document has a hidden associated topic distribution that is frequently used to efficiently retrieve matching documents. A Support Vector Machine [24] is an algorithm which, given a set of training examples, each marked as belonging to one of two categories, generates a supervised learning model that assigns new examples to one of those categories.

Our approach to the classification problem is divided in three steps: first, we discover the different hidden topics of the various tweets of the train set; next, we automatically aggregate the combinations of those subjects as relevant or irrelevant to the *seed*; and finally, we train a binary classifier that considers the labels of the tweets and their topic distributions to assign labels of *relevant* or *irrelevant* to new tweets.

For the first step, we consider that every preprocessed tweet from our data set is a single document and build a Document Term Matrix (DTM) based on the collected corpus. This DTM is then used to train a topic model, determining the topic distribution associated with each individual tweet.

For the second step, we need to divide the tweets in two segments according to their relevance to the original topic. For this task, we inspect the DTM and label each document of the data set as *relevant* if it contains the *seed* (preceded or not by the octothorp) or *irrelevant* otherwise. This classification is not perfect in the sense that some documents related to the topic may be incorrectly labeled for not containing the exact *seed*. However, from our experience, this tweets usually represent a negligible percentage of the train set and this labeling process is what enables the training of our classifier without any supervision.

On the final step, we associate the topic distributions of each tweet with their corresponding label, thus aggregating all the combinations of latent topics available in the train set in two different classes. After collecting a train sample, the support vector machine is trained to classify new tweets as *relevant* or *irrelevant* based on their corresponding topic distributions.

The pseudo-code representation of the method described above to train the classifier is shown in Algorithm 2 .

Algorithm 2: Training the Hashtag Classifier.

Input: *AllTweets*, *seed*, *k*
Dtm ← document term matrix of *AllTweets*
LdaModel ← LDA model of *Dtm* with *k* topics
RelevantDistributions ← all topic distributions from *Tweets* containing *seed*
RelevantDistributions$*Class* ← relevant
IrrelevantDistributions ← all topic distributions from *Tweets* not containing *seed*
IrrelevantDistributions$*Class* ← irrelevant
AllDistributions ← *RelevantDistributions* ∪ *IrrelevantDistributions*
Classifier ← SVM model to predict *class* from *AllDistributions*
return *Classifier*

The functions used to build this classifier in R were `LDA()` from the package `topicmodels` [9], to build the topic model based on Latent Dirichlet Allocation, and `SVM()` from the R package `e1071` [16], to train the support vector machine.

3.3. Identifying relevant hashtags

The final stage of the process is similar to the first one for data collection, with some added procedures for filtering each tweet, in place, as relevant or irrelevant to the given topic, using the classifier.

After training the classifier, we start by employing the Twitter Search API to query Twitter and collect a combination of the most recent and the most popular tweets containing the *seed*. However, this time we want to remove the irrelevant tweets from the set before inspecting them for hashtag retrieval.

We consider the retrieved assortment of tweets as a corpus of documents and use it to generate a new Document Term Matrix with exactly the same terms as the DTM used to train the classifier. Through this new DTM, the posterior probabilities of the topics for each document are calculated, that is, the topic distribution of each tweet is determined according to the topic model previously used for training the classifier. While the process of calculating this posterior probabilities may seem complex at first, software packages for topic modeling already include an implementation of the required algorithms based on Gibbs sampling.

After calculating the topic distributions of the corpus, the classifier is employed for labeling each tweet as *relevant* or *irrelevant* and subsequently discard all the irrelevant ones from the group. The hashtags from the tweets identified as *relevant*, whose frequency is above a threshold (determined empirically as in Section 4.1.3), are added to a queue and stored in an array. The process is repeated with the new search terms in the queue until it becomes empty or a limit of iterations is reached. At the end of the execution, the mentioned array contains a list of hashtags currently relevant to the explored *seed*.

Every new tweet collected during the execution of this step is also stored and added to the corpus used to train the classifier in the next executions. The idea is the classifier should provide more accurate results each time TORHID is used to collect hashtags, since we add more data from the previous iteration.

The pseudo-code representation of the method described above to collect the relevant hashtags is shown in Algorithm 3 .

Algorithm 3: Identifying the Relevant Hashtags.

Input: *seed*, *max.searches*, *threshold*, *Dtm*, *LdaModel*, *Classifier*
HashtagQueue ← initiate queue with *seed*
AllTweets ← ∅
RelevantHashtags ← ∅
while *HashtagQueue not empty* ***and*** *max.searches* > 0 **do**
  CurrentHT ← dequeue *HashtagQueue*
  NewTweets ← set of retrieved tweets containing *CurrentHT*
  AllTweets ← *AllTweets* ∪ *NewTweets*
  NewTweets ← preprocessed *NewTweets*
  DtmTweets ← document term matrix of *NewTweets* (with terms of *Dtm*)
  TopicTweets ← topic distributions of *DtmTweets* (with topics from *LdaModel*)
  RelevantTweets ← tweets labeled *relevant* by Classifier
  NewHashtags ← hashtags in *RelevantTweets* more than *threshold* times
  HashtagQueue ← *HashtagQueue* ∪ *NewHashtags*
  RelevantHashtags ← *RelevantHashtags* ∪ *NewHashtags*
  max.searches ← max.*searches* − 1
end
store *AllTweets*
return *RelevantHashtags*

In addition to the ones introduced in previous steps, the function used to implement this method for collecting hashtags relevant for a *seed* in R was *Posterior* from the package `topicmodels` [9], to calculate the topic distributions of the new tweets according to the provided topic model based on Latent Dirichlet Allocation.

3.4. Relations of the hashtags

Now that we have the final collection of hashtags, we can inspect the relations among them. For this purpose, we search every hashtag of the list on Twitter again and use a matrix to record the number of times each of the other hashtags appear in the fetched tweets for each query.

The pseudo-code representation of the method described above to weight the relations among the retrieved hashtags is shown in Algorithm 4 .

Algorithm 4: Weighting the Relations of the Hashtags.

```
Input: HashtagQueue
Weights ← square matrix with number of HashtagQueue
while HashtagQueue not empty do
    CurrentHT ← dequeue HashtagQueue
    NewTweets ← set of retrieved tweets containing CurrentHT
    NewHashtags ← hashtags in NewTweets
    index ← 1
    while index < number of HashtagQueue do
        Weights[CurrentHT][index] ← number of times the hashtag in index appears in NewTweets
        index ← index + 1
    end
end
return Weights
```

For a graphical representation of the relationships of the hashtags, we can consider the retrieved matrix as an adjacency matrix and use the R package `igraph` [3] to draw a directed graph where the vertices are the hashtags and the weights of the edges are the values in the matrix.

4. Evaluation

Considering that English words are often ambiguous and that some hashtags can be simply acronyms or abbreviations, it is a non-trivial task to accurately measure how much a hashtag is relevant or not to a given topic, especially without providing any context. On the other hand, as far as we know, an established general method to evaluate topic models is not currently available. Considering that our goal is to define relevance as recognized by human beings, we split the evaluation of our methodology in two different stages: first, we test the performance of the classifier by experimenting with multiple combinations of the parameters and by comparing the results to a manual labeling; later, we examine the network of hashtags returned by TORHID and analyze it in a qualitative way, comparing its performance with a commercial tool.

4.1. Quantitative evaluation

For the quantitative evaluation, we have chosen three topics to test: *#preterm, #cerebralPalsy* and *#gunControl*. While one of the advantages of TORHID is the collection of tweets in real time directly from Twitter, and consequentially not requiring the maintenance of large data sets, this approach becomes inconvenient when experimenting with different parameters and thresholds since the results would naturally variate over time. As such, for this analysis we employed our Data Collection tool to construct three data sets capable of representing segments of Twitter somewhat relevant to our *seeds* at a particular time, with the intention of using them to later simulate the direct searches.

For each *seed*, over the course of two days, we employed Algorithm 1 to search 1000 hashtags (without minimum frequency restrictions) and collected up to 1000 tweets for each of those hashtags. Since the Twitter Search API imposes limitations for the age of the retrieved tweets, fetching exactly 1000 is unlikely for most hashtags. For this reason the three final data sets contained slightly more than 500.000 tweets each.

4.1.1. SVM classifier

The effectiveness and versatility of classifiers based on support vector machines is dependent of applying the correct kernels and respective variables for each circumstance. However, the conventional method of determining the best parameters for training a SVM is still trial and error. In this subsection, we experiment with combinations of different values for the kernel, the γ and the cost, as well as with the removal of the *seed* terms from the respective document term matrices.

We started by sampling the simulation data sets to generate train sets and test sets, with 30.000 tweets and 1.500 tweets respectively, for each topic. We proceed by manually labeling the tweets of the test sets into *relevant* or *irrelevant* to the corresponding *seeds*. Since analyzing tweets according to the topics is subjective, this task was performed by two persons independently. While neither of these persons was an expert in the explored topic, searching about it (or any retrieved tweet) online was encouraged. The classifications were subsequently compared and when the two persons did not agree in a particular label, they discussed until a consensus was reached.

We selected four different kernels (*linear, polynomial, radial* and *sigmoid*) to test the support vector machine and three different values for experimenting with the γ (1, 10, 100) and the cost (1, 10, 100). Considering that γ and cost could be mutually dependent, we decided to test all combinations of the values. For each *seed*, we used LDA to build a model according to the corpus, with 25 hidden topics, and used it to train the SVM for every combination of kernels and variables (except for the linear type of kernel that doesn't require γ and cost). The classifications of the tweets were later compared with the manual labeling done in previous stages for calculating the percentages of correct and incorrect classifications.

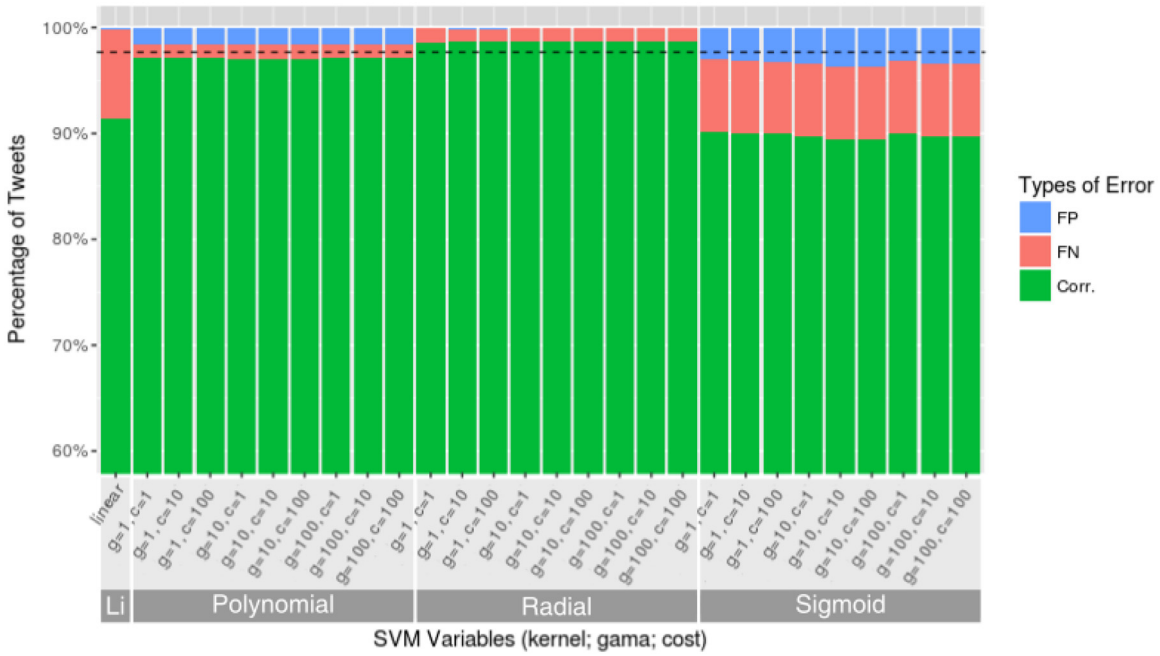

Fig. 3. Percentage of types of error with different SVM variables for the Hashtag *#Preterm*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

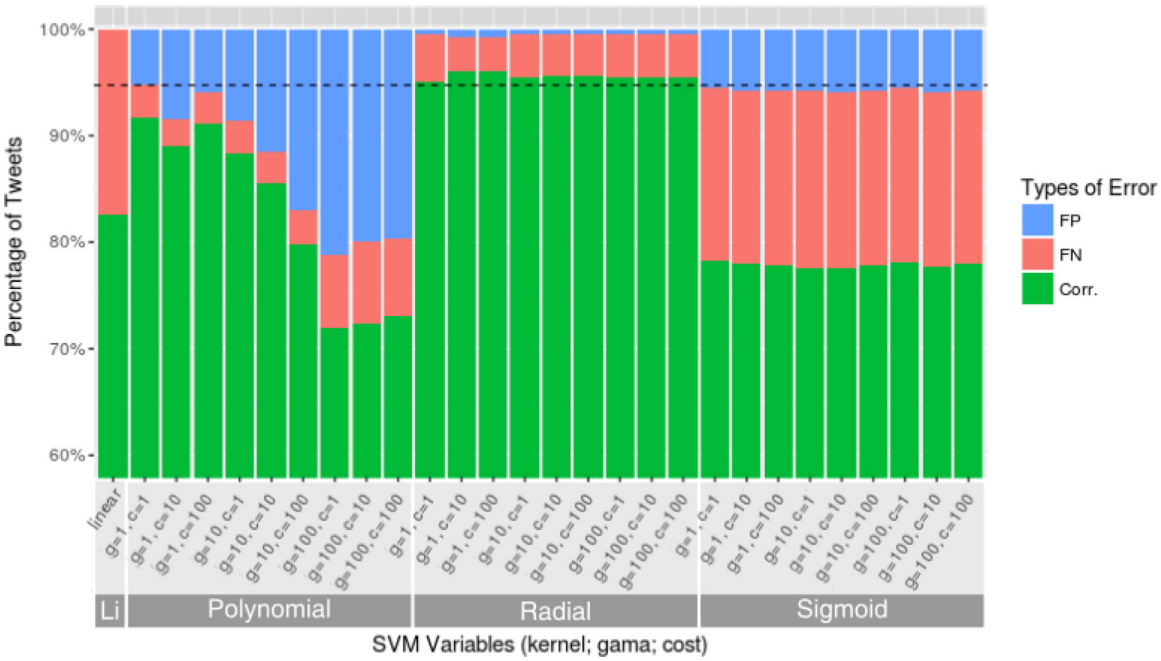

Fig. 4. Percentage of types of error with different SVM variables for the Hashtag *#CerebralPalsy*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For a comparison metric, we developed a simple classifier that works in the following manner: every tweet containing the *seed* is labeled as *relevant* and every tweet not containing the *seed* is labeled as *irrelevant*. This classifications were also compared with the manual labels for calculating the percentages of right and wrong classifications. The graphical summary of the results for the experiments with the different variables are described in Fig. 3 for the *seed #Preterm*, in Fig. 4 for the *seed #CerebralPalsy* and in Fig. 5 for the *seed #GunControl*, with the dashed black lines representing the threshold of the

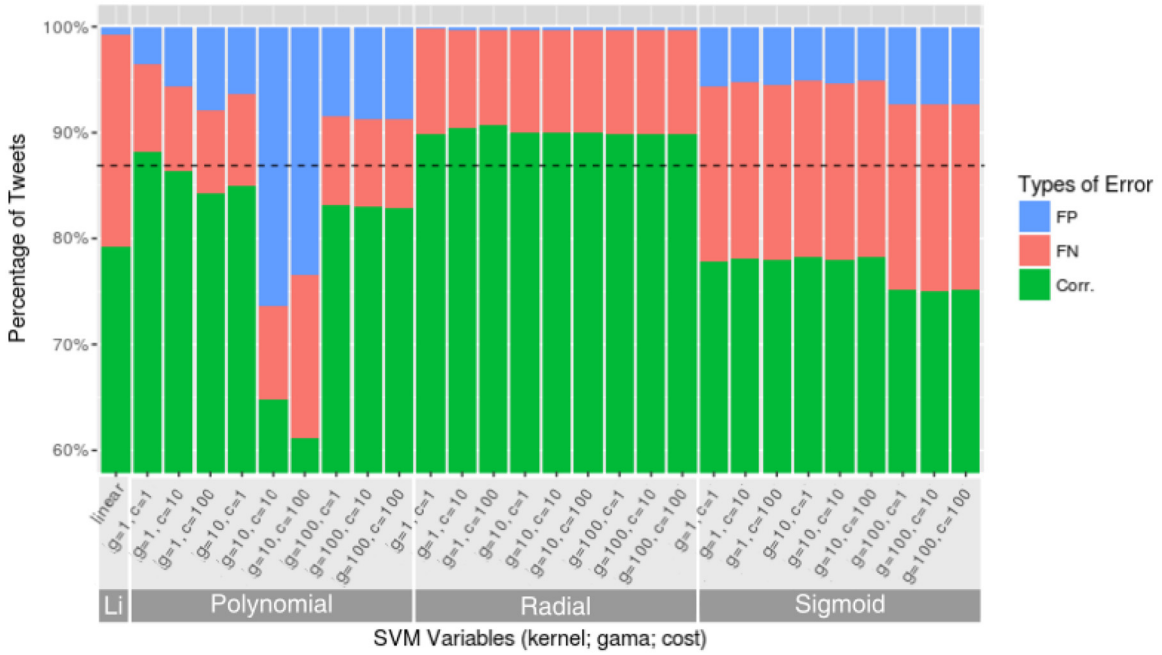

Fig. 5. Percentage of types of error with different SVM variables for the Hashtag *#GunControl*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

alternative algorithm and the colors green, red and blue representing the percentage of correct classifications, false negatives and false positives respectively.

According to our tests, it seems apparent that a radial type kernel performs better for our use case of finding relevant tweets than the other types of kernel, since it generally provides more accurate and more consistent results, always surpassing the alternative of only considering the presence of the *seed* in the text. On the other hand, the variations in the results of experimenting with different values for γ and cost are usually negligible, particularly for the radial kernel where using a combination of $\gamma = 1$ and $cost = 100$ consistently improved the number of correct classifications, but by an insignificant margin. Finally, the percentage of false positives is consistently very low, which results in great precision values. This is important for the overall reliability of the algorithm since it is expected to translate into a smaller probability of returning hashtags unrelated to the seed.

A shortcoming of our approach that is not expressed in this experiment is the amount of time required for training. Depending on the size of the training data, TORHID may need up to an hour to train the topic model and the support vector machine. When using the right kernel, the classifications provided by TORHID were consistently superior to the alternative of only considering the presence of the *seed* in the text. However, this alternative method was always ready for classifying the tweets faster than TORHID, since it does not require a previous step for model training. Furthermore, since the use of a train set would be pointless for this method, the procedure of data collection can be ignored, additionally saving up to several days. For this reason, employing the simple classifier method could be sufficient in cases where a faster execution time would be prioritized over the better results achieved by TORHID.

In the next step, we test the classification of each tweet dependence on the presence or not of the *seed*, that is, if removing the *seed* term from a relevant tweet would make the classifier incorrectly label it as *irrelevant* instead. It is expected that, by removing the *seed*, the classifier may become less biased towards it. It is important to notice that, since the corpus was preprocessed, all the octothorps were removed, rendering the hashtags indistinguishable from the regular words which comprise them.

For this task, we first edited the document term matrices of the train and the test sets of each topic and removed the term corresponding to the stemmed *seeds* and proceeded to repeat the previous tests with exactly the same kernels and variables combinations. The graphical summary of these results are described in Fig. 6 for the *seed #Preterm*, in Fig. 7 for the *seed #CerebralPalsy* and in Fig. 8 for the *seed #GunControl*, once again with the dashed black lines representing the threshold of the alternative algorithm that only considers the presence of the *seed* (obviously not removed) and the colors green, red and blue representing the percentage of correct classifications, false negatives and false positives respectively.

It is immediately noticeable that the support vector machines without access to the *seeds* provided inferior classifications, not only when compared to previous results where the *seeds* were maintained in the document term matrices, but also when compared to the alternative algorithm that only considers the presence of the *seed*. This fact is probably due to the higher chance of the model to classify most tweets with the seed hashtag as positive. Despite the lower percentage of

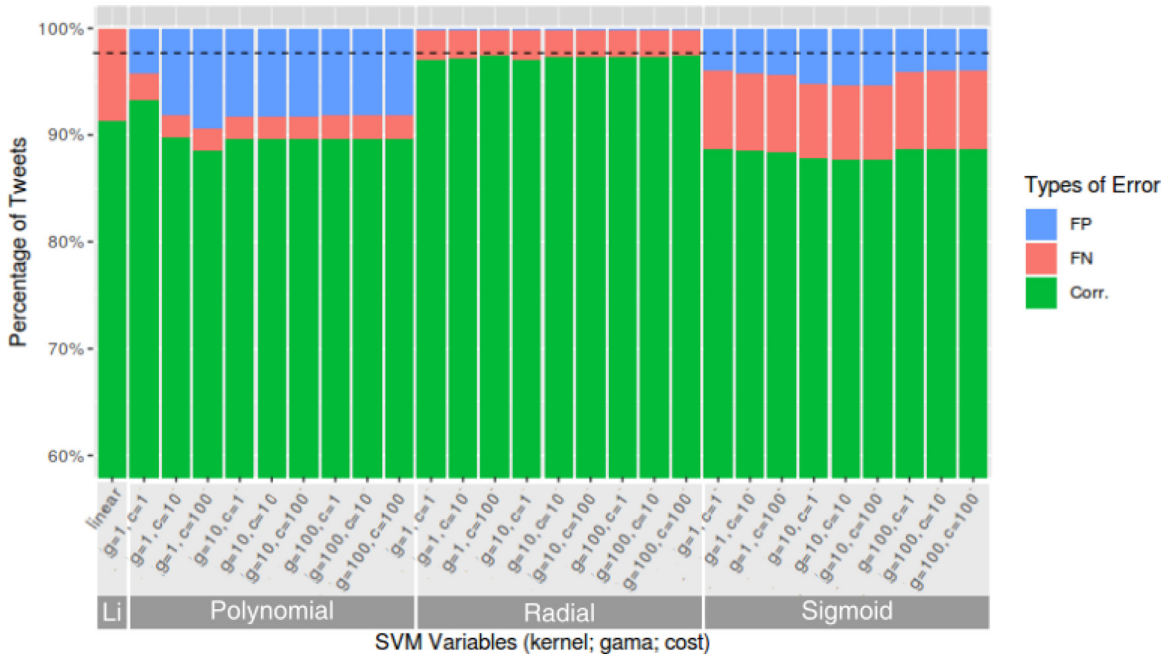

Fig. 6. Results for different combinations of SVM variables without *Seed* for the Hashtag *#Preterm*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

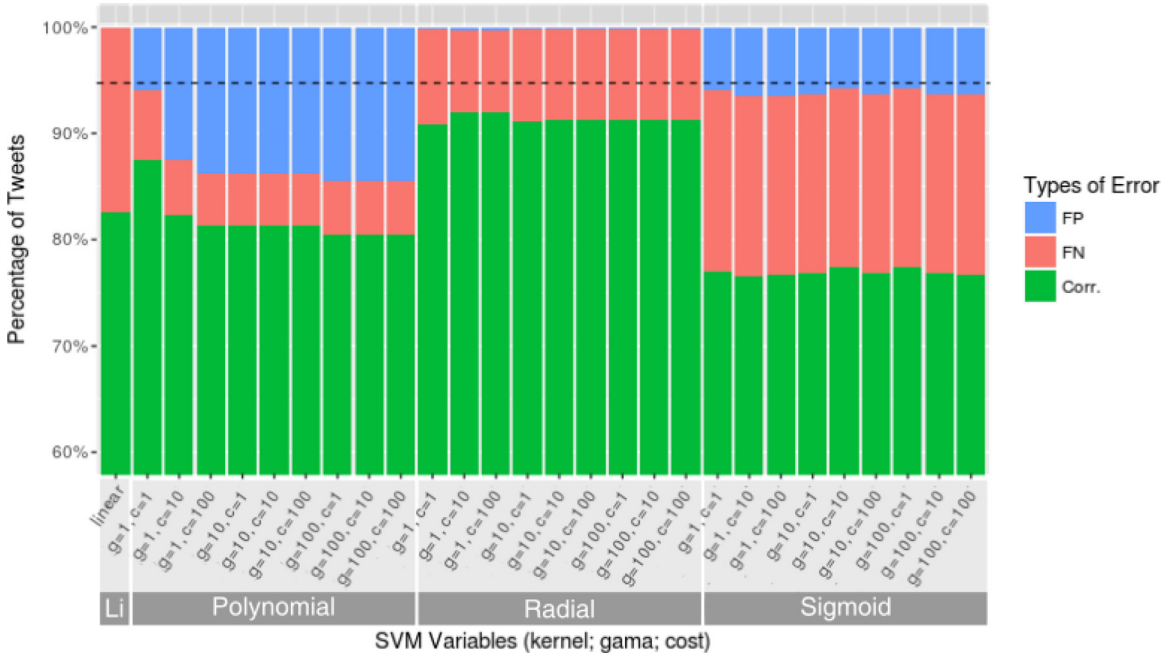

Fig. 7. Results for different combinations of SVM variables without *Seed* for the Hashtag *#CerebralPalsy*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

correct classifications, the results were consistent with the previous experiments, with the radial type kernel proving to be more accurate than the alternatives and the different γ-cost combinations failed to significantly impact the classifications for most cases.

Overall, the results from these experiments show that a radial type kernel with $\gamma = 1$ and $cost = 100$ will generally achieve the best classifications and that removing the *seed* from the document term matrix results in a slight deterioration of the results.

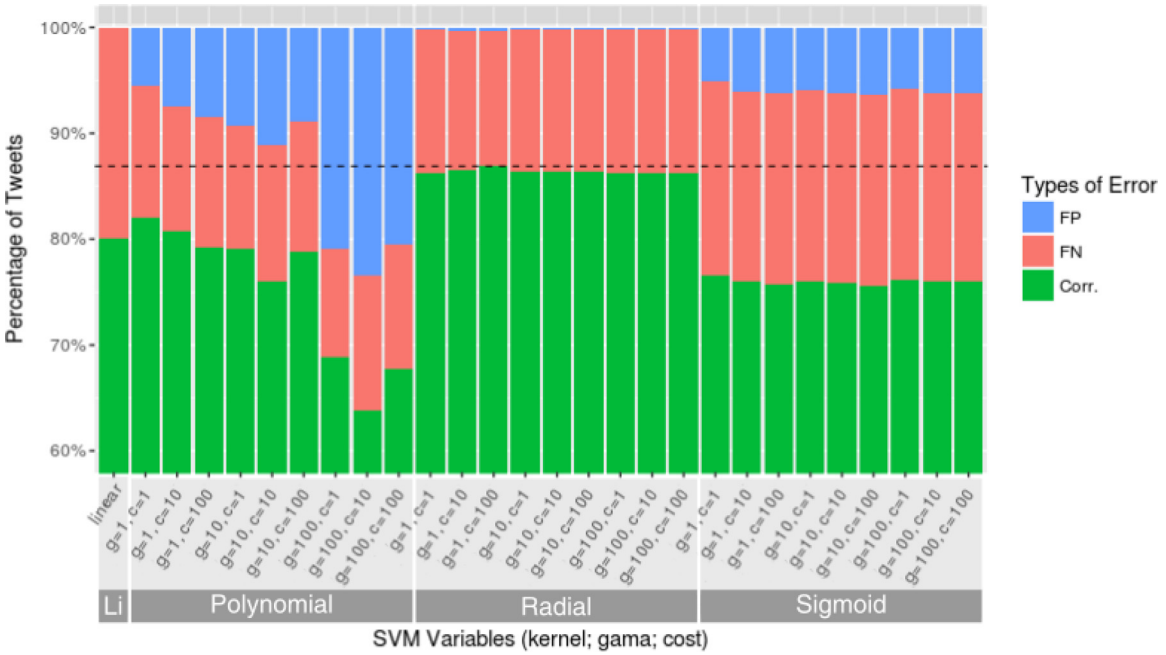

Fig. 8. Results for different combinations of SVM variables without *Seed* for the Hashtag *#GunControl*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.1.2. LDA model

Since the generated topic model based on Latent Dirichlet Allocation influences the train data of the support vector machine, it is also important to be tested. In this subsection, we experiment with different values for the number of hidden topics and with different sizes of samples for the training sets.

We started by sampling the simulation data sets to generate different train sets with 5.000, 10.000, 30.000 and 50.000 tweets for each *seed*. We also decided to use the same test sets as used in the previous tests to maintain consistency among the experiments.

For each train set, we employed Latent Dirichlet Allocation to build multiple topic models according to the respective corpus, using different numbers of latent topics (5, 10, 25, 50, 75). Next, the topic models were used to train the support vector machines, with a radial kernel and associated $\gamma = 1$ and $cost = 100$, and classify the tweets from the test set. As before, the classifications of the tweets were compared with the manual labels and the percentages of correct and incorrect classifications were calculated.

Next, the results were grouped according to the size of the train sets (5.000, 10.000, 30.000 and 50.000 tweets) and the averages of the values were calculated accordingly. The graphical summary of the results of modeling the LDA with different sized samples is described in Fig. 9 for the *seeds #Preterm, #CerebralPalsy* and *#GunControl* with the colors green, red and blue representing the percentage of correct classifications, false negatives and false positives respectively.

The results are consistent across the three different *seeds*: the proportion of correct classifications increases with the size of the train sets. However, the increase of performance seems to be logarithmic instead of linear, being significantly lower when expanding the train set from 30.000 tweets to 50.000, compared to expanding the train set from 5.000 tweets to 10.000. Nonetheless, this may be due to the closed approach of our tests and bigger train sets may prove to be beneficial in long term solutions where an higher ratio of relevant tweets could be maintained and older tweets would be mixed with new ones over time.

The results were then regrouped according to the number of hidden topics (5, 10, 25, 50, 75) and the averages of the values were again calculated accordingly. The number of topics is usualy a trade-off between a too general model that has few topics and a very specific model which needs a lot of training data to avoid overfitting. The graphical summary of the results of modeling the LDA with different sized samples is depicted in Fig. 10 for the *seeds #Preterm, #CerebralPalsy* and *#GunControl* with the colors green, red and blue representing the percentage of correct classifications, false negatives and false positives respectively.

Again, the results are consistent across the three different *seeds*: the proportion of correct classifications increases with the number of latent topics until 25, decreasing in again at 75 hidden topics. In this manner, the results suggest an ideal number of topics between 25 and 50 to generate the LDA model. It is expected that smaller train sets would benefit from less latent topics and larger train sets from more.

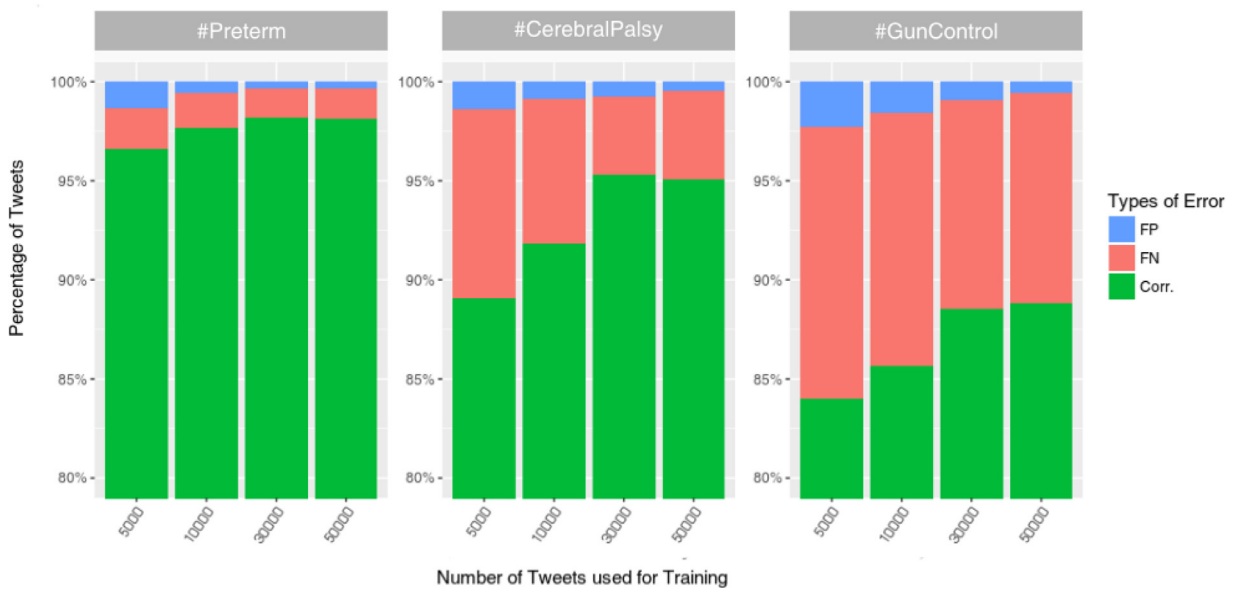

Fig. 9. Average of the percentages of the types of error for different sizes of train sets and different *seeds*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

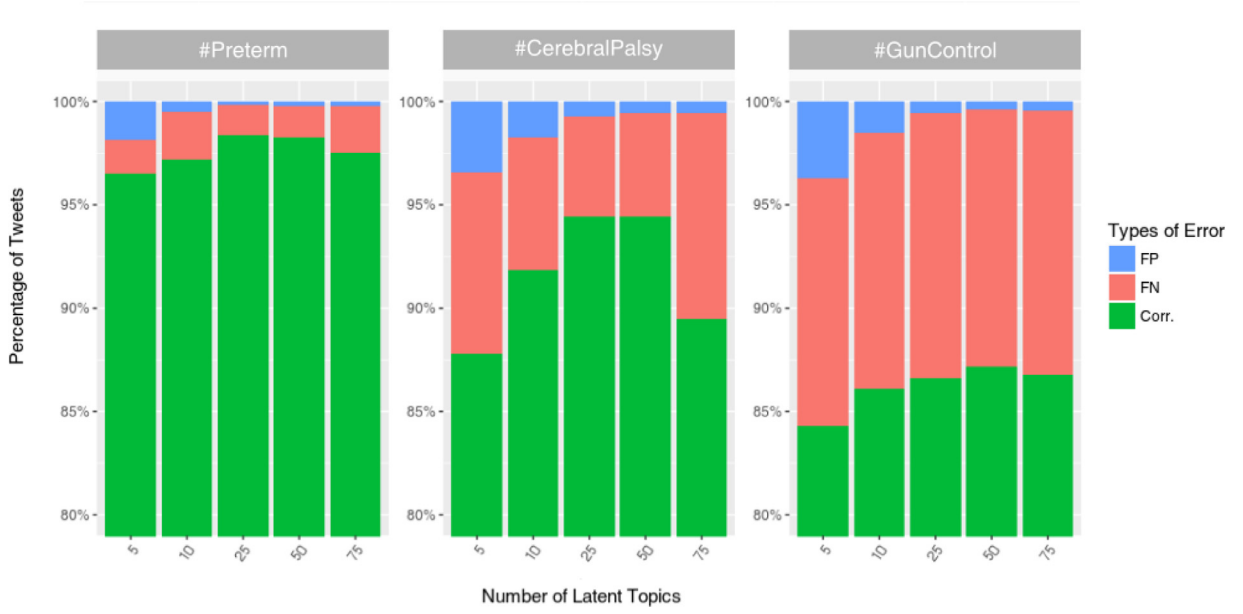

Fig. 10. Average of the percentages of the types of error for different numbers of latent topics and different *seeds*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Generally, the results from these experiments show that the performance of the classifier increases with the amount of training data and, in this particular case, a train set with 30.000 is already enough for achieving positive results. On the other hand, the best number of latent topics seems to be approximately between 25 and 50 in every case.

4.1.3. Search variables

After determining the best parameter for the classifier, we proceed to test the variables of another key component of our solution: the search mechanism. While the classifier is important to filter unrelated hashtags, the search element of the algorithm is responsible for the actual collection of the tweets, as well as a simple selection based on frequency thresholds. In this subsection, we experiment with different values for the number of search iterations, the quantity of tweets collected in each search and the minimum frequency for hashtags being considered relevant.

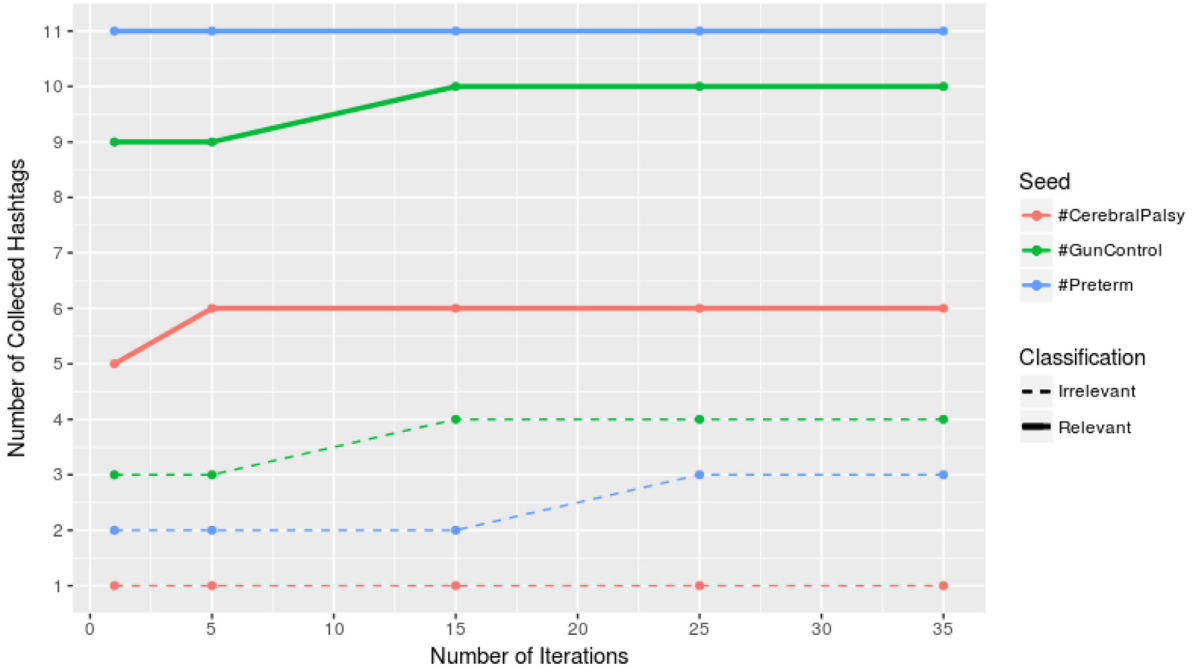

Fig. 11. Number of hashtags collected with different number of iterations for different *seeds*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We started by training the classifiers with the best performing parameters according to the previous tests: for each *seed*, the same train sets with 30.000 tweets were employed to generate the respective topic models with 25 hidden topics and the support vector machines were trained with a radial kernel, $\gamma = 1$ and $cost = 100$.

Next, we simulated the execution of TORHID in a controlled scenario by running multiple searches for each *seed* in the respective data sets. The tool was configured to collect a maximum of 250 tweets for each search and the threshold for minimum hashtag frequency was set to 3, while we experimented with different numbers of iterations: 1, 5, 15, 25 and 35. The hashtags retrieved in each execution were later manually labeled as relevant or irrelevant to the respective *seed* by four different people independently and compared to reach a consensus. It was established that, this time, too general hashtags would be labeled *irrelevant* (despite generality also being somewhat subjective). The results of searching with a different number of iterations are summarized in Fig. 11 for the *seeds #Preterm* represented in blue, *#CerebralPalsy* in red and *#GunControl* in green, while the solid lines represent the relevant hashtags and the dashed lines represent the irrelevant ones.

While the total number varies, the amount of relevant hashtags is always superior to the number of irrelevant hashtags. Also, raising the number of iterations in some cases may increase the number of hashtags retrieved. However, this pattern occurs at a very small rate and stops increasing at a peak with 25 iterations. This is probably due to the most relevant hashtags being closer to the *seed* in the relations network and the algorithm never collected more than 14 hashtags in total. In this case, we consider that 15 iterations provided the most satisfactory results, being the lower number to collect the maximum relevant hashtags in total.

We proceeded to investigate the impact of the number of tweets collected by each search. Using the same classifiers and data sets as in the previous test, we repeated the simulations with 15 search iterations and a fixed threshold of a minimum frequency of 3 times for each hashtags, while experimenting with the number of collected tweets: 50, 150, 250 and 350. The hashtags retrieved in each execution were again manually labeled and compared through the same process as before. The results of searching different numbers of tweets per iteration are summarized in Fig. 12 for the *seeds #Preterm* represented in blue, *#CerebralPalsy* in red and *#GunControl* in green, while the solid lines represent the relevant hashtags and the dashed lines represent the irrelevant ones.

As before, the number of relevant hashtags remains superior to the number of irrelevant ones. Nonetheless, it is noticeable that the number of collected hashtags regularly increases with the number of collected tweets. This increase is expected since the hashtags are contained in the tweets and thus collecting more different tweets corresponds naturally to collecting more different hashtags. While the number of hashtags for *#CerebralPalsy* and *#GunControl* keeps increasing almost linearly, for *#Preterm* it stops when it reaches 150 tweets, what may be a sign of scarceness or high similarity in the tweets collected for this *seed*.

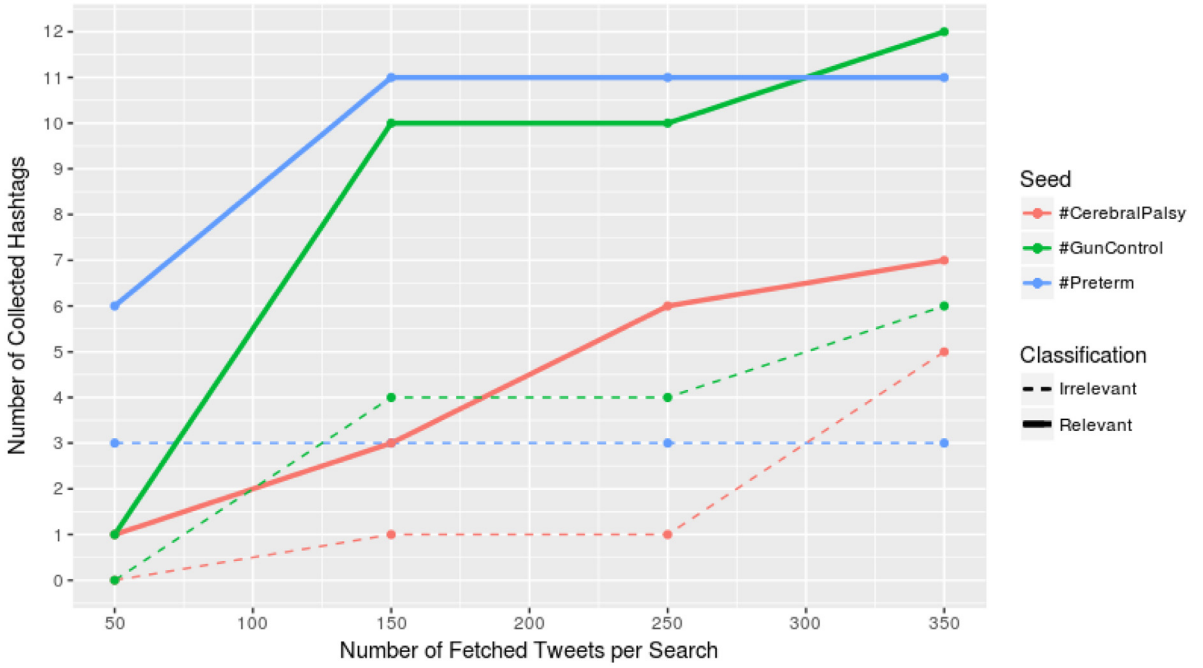

Fig. 12. Number of hashtags collected with different number of tweets per iterations for different *seeds*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Finally, we examine the influence of the hashtag frequency threshold. Once again, we employ the same classifiers and data sets and repeat the simulations, this time with 15 search iterations collecting 250 tweets, while experimenting different frequency thresholds: 1, 3, 5, 7, 9 and 11. The hashtags retrieved in each execution were also manually labeled and compared through the same process as before. The results of varying the minimum frequency for considering hashtags relevant are summarized in Fig. 13 for the *seeds #Preterm* represented in blue, *#CerebralPalsy* in red and *#GunControl* in green, while the solid lines represent the relevant hashtags and the dashed lines represent the irrelevant ones.

Considering the proportions of relevant and irrelevant hashtags, the results remain consistent with the previous ones. However, this time we experience an exponential decay while increasing the frequency threshold. This results are expected since tweets are limited to 280 characters, forcing users to select fewer hashtags. Furthermore, our tests suggest that increasing the frequency threshold may filter the hashtags by degree of relevance to the *seed* since setting its value to 9 resulted in the retrieval of only relevant hashtags, albeit in smaller quantity. With a threshold large enough, only the *seed* would be returned.

In addition, commonly to all experiments in this section, the number of hashtags collected for *#Preterm* and *#GunControl* is always higher than for the hashtag *#CerebralPalsy*. This may be due to cerebral palsy being a sensible theme, what may result in less discussions and shares on Twitter, leading to the tweets containing the hashtag being scarcer and similar to each other. On the other hand, gun control is currently a major discussion topic in the United States of America and preterm births, while not a popular conversation theme for the average person, performed well in our tests due to charity campaigns ongoing when the tweets were collected.

4.2. Qualitative evaluation and comparison

The method to qualitatively evaluate TORHID consists in the comparison of its returned hashtags with the alternatives depicted in the related work section. Despite most of the described approaches being different from ours in their purpose and operation, the main goal of *Hashtagify* resembles ours and the results of both approaches can be compared. As previously mentioned in Chapter 2, *Hashtagify* is a payed web service whose purpose is to allow its clients to search hashtags' relations and influencers in order to improve their social media strategy. The free version limits the user to search for a hashtag and retrieve a word cloud with the 10 most related hashtags. Other features are advertised in beta for clients with a payed subscription, but are currently unavailable.

Since *Hashtagify* constitutes a business, its source code and algorithms are closed source and so, unlike with the previously proposed approach, we were forced to employ a black-box testing method to evaluate and compare their service with TORHID. The evaluation process was the following: first, we randomly selected 20 different hashtags from arbitrary topics, including single words, concatenations of words and acronyms representing different subjects from currently trending matters to more obscure themes. Next, for each topic, we used TORHID to collect the corresponding relevant hashtags and

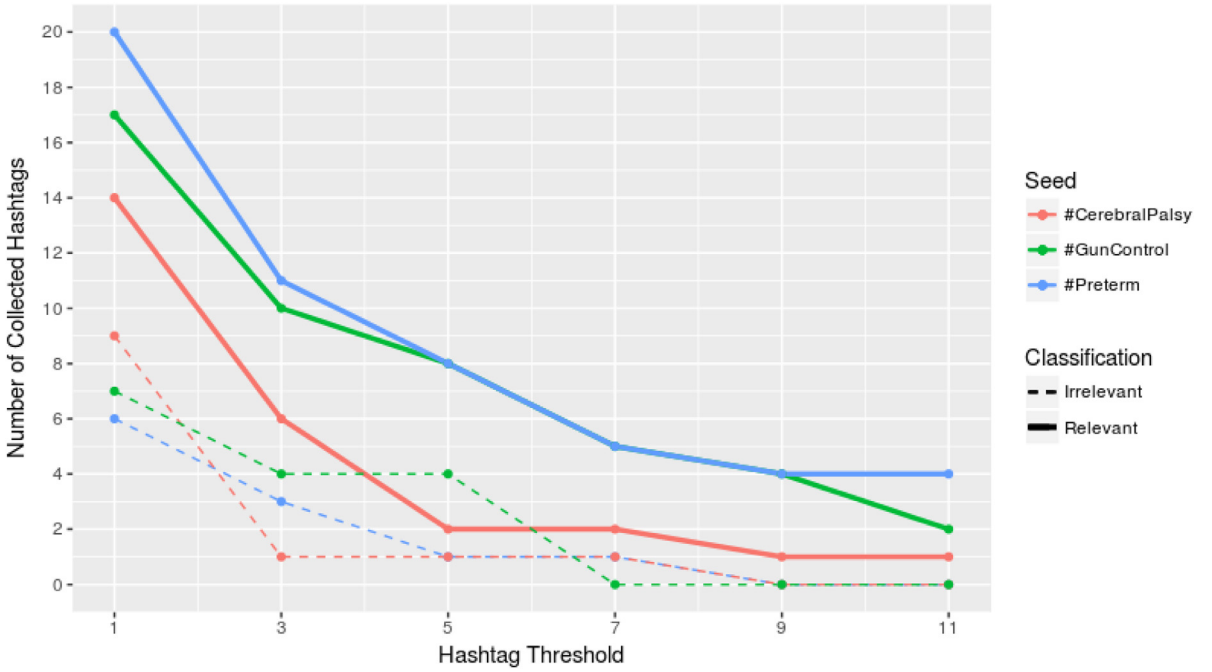

Fig. 13. Number of hashtags collected with different frequency thresholds for different *seeds*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Seeds explored for the qualitative evaluation.

Alzheimer	Eurovision	Lisbon	NBA	Sunset
Apple	Gaza	Microsoft	Nokia	Tumor
Bitcoin	Guncontrol	Moneyconf	Preterm	WWE
Cerebralpalsy	InfinityWar	MLB	Spring	Workout

Table 2
Comparison of the results collected by TORHID and *Hashtagify*.

	TORHID		**Hashtagify**	
	Arithmetic Mean	σ	**Arithmetic Mean**	σ
Relevant	201,75	9,639	89,25	4,573
Irrelevant	33,75	11,529	31,5	11,091
Neutral	64,5	12,557	28,25	7,805
Total	300	N/A	149	N/A
Relevant (%)	67,25%	0,032	59,90%	0,031
Irrelevant (%)	11,25%	0,038	21,14%	0,074
Neutral (%)	21,50%	0,042	18,96%	0,052

represent their relations in a graph. At the same time, we searched the same hashtag in *Hashtagify* and stored the word cloud. The hashtags retrieved by both approaches were collected for later categorization. After, instructions were given to four people to independently classify each hashtag of the obtained network and word cloud as *relevant, irrelevant* or *neutral* to the main topic. While the expertise of these people varied for each different topic, searching the meaning or context of a hashtag was encouraged. In case of a tie, the four people argued until a consensus was reached. Table 1 presents the *seeds* explored in this test. In total, 449 hashtags were classified and Table 2 compares the mean and standard deviation of the results from TORHID and *Hashtagify*.

Despite the percentage of *neutral* hashtags being very proximate for both tools, our hashtags were classified as 7.35% more *relevant* and 9.89% less *irrelevant* than the ones from *Hashtagify*. Usually, those percentage differences would not translate to a tremendous improvement on themselves. However, *Hashtagify* rarely succeeded in retrieving 10 hashtags as selected (only accomplished that goal three times out of twenty) while TORHID collected more than 10 hashtags in the majority of times. On average, our technique collected 15 hashtags for each topic, out of which 10.09 were considered relevant, while *Hashtagify* retrieved on average 7.45 hashtags, out of which 4.46 were considered relevant. Both results were satisfactory,

Table 3
Comparison of the hashtags collected for the topic *#Preterm*.

#Preterm		
Classification	**TORHID**	**Hashtagify**
Relevant	prematurebaby, prematurity, premmies	borntoosoon, premature, pretermbirth
Contextual	countrychildmag, kangaroomothercare	nicu
Neutral	babies, newborn	birth, infants, pregnancy
Irrelevant	–	bluecj, wellnesswed
Total	7	9

Table 4
Comparison of the hashtags collected for the topic *#CerebralPalsy*.

#CerebralPalsy		
Classification	**TORHID**	**Hashtagify**
Relevant	cerebralpalsyawerenessmonth, disabilityrights	cp, disability, disabled
Contextual	westham	hyperbaric
Neutral	autism	autism
Irrelevant	writer	inspiration
Total	5	6

Table 5
Comparison of the hashtags collected for the topic *#GunControl*.

#GunControl		
Classification	**TORHID**	**Hashtagify**
Relevant	assaultweaponsban, boycottthenra, guncontrolnever, gunsense	floridaschoolshooting, nra
Contextual	boycott, marchforo, neveragain, oregon, parkland, parklandstudents, wecallbs	neveragain, parklandstudentsspeak
Neutral	–	–
Irrelevant	–	–
Total	11	4

particularly when considering the low ratio of false positives, that is, irrelevant hashtags. However, it was considered that, overall, TORHID outperformed *Hashtagify*.

One objective of TORHID not directly examined with the evaluation process described above is if the hashtags returned are always relevant to the topic or only during a particular temporal length and if that interval of time is present or past. This factor is important since the recency of the collected hashtags may be decisive in some cases. Therefore, we followed by experimenting a variation of the previous evaluation with some extra considerations but in much lower extent because of the added difficulty in accurate classification. The adjustments to the evaluation procedure described above were that, this time, only three arbitrary topics were chosen and only two people independently classified the hashtag. Also, the classes of the hashtags remained: *relevant* if it immediately had direct bearing on the topic (or it was a subtopic); *neutral* if it was connected to the topic but too general (or it was a supertopic); *irrelevant* if it was completely unrelated to the given topic; and a new *contextual* class was added if the hashtag seemed irrelevant at first sight but after searching for the subject it was found to be relevant (at least for that particular given time). When the two people did not agree, they discussed until a consensus was reached. The hashtags collected by each method, as well as their classifications are described side by side in Table 3 for the topic *#Preterm*, in Table 4 for the topic *#CerebralPalsy* and Table 5 for the topic *#GunControl*.

At first glance, both approaches seemed to produce somewhat similar results. However, after a closer inspection, it is noticeable that only two pairs, from a total of the forty three hashtags returned (considering the three topics combined), were exactly the same for the two methods. This phenomenon is probably due to the way TORHID works with more recent data compared to *Hashtagify* which relies on a large longstanding database maintained by the company.

When searching the *seeds #Preterm* and *#CerebralPalsy*, TORHID returned less hashtags than *Hashtagify*. This may be due to subjects related to medical conditions being too intimate for being shared (personal experiences) or too specialized for becoming popular (medical findings), thus benefiting from long term data. However, when discarding the irrelevant hashtags, TORHID stays on pair with *Hashtagify*. Hashtags like *#wellnesswed, #writer* and *#inspiration* expressed contexts too general for being considered related to the topics and *#bluecj* was an obscure hashtag, to the point of a web search on the term getting almost no results.

On the other hand, when searching for *#GunControl* our results were significantly better than *Hashtagify*'s. While none of the approaches retrieved any irrelevant or even neutral hashtags, TORHID collected more than the double of the hashtags retrieved by *Hashtagify*, although *#neveragain* was a common hashtag to both systems. This is probably due to the character

of the topic being currently very popular and publicly debated in the United States of America (the hashtags were collected a few days after the Stoneman Douglas High School shooting [11]), what benefits a short term and more embracing search.

Contextual results are particularly important for the method we propose, since they are composed of hashtags which either are not immediately obvious to the average person or whose relevancy to the topic is variable within time and context, i.e. hashtags that are recent. This means that, while *#Westham* is an association football team, at that point in time they were relevant to the topic *#CerebralPalsy* because of their campaign during the Cerebral Palsy Awareness Month to gather funds to help support people with the condition. Meanwhile, *#Parkland* became relevant to the topic *#GunControl* because of the infamous school shooting which toke place in the city days before we searched for relevant hashtags. Both of these hashtags, as well as the majority of the ones classified as contextual, will probably cease to be relevant to this topics in the future but are pertinent right now, so they are entitled to a different classification. Considering that TORHID searches Twitter for the latest tweets every time, it has an advantage in returning contextual hashtags when compared to methods that rely on longstanding databases, like *Hashtagify* which take longer to be updated.

Overall, we consider that our results outperform the ones provided by *Hashtagify*, with a similar number of relevant hashtags and a higher number of contextual ones. Our approach also has the advantages of requiring a smaller database and collecting more up to date hashtags, while also returning a network with the relations among each other. On the other hand, the dependency of *Hashtagify* in a extensive and longstanding database brings some advantages, particularly in the short time required to answer a query to their database (few seconds), while TORHID may need up to an hour to train the classifier, in addition to the necessary time to collect the train set. Since the hashtags retrieved by both approaches were generally different, using both methods when searching Twitter for hashtags relevant to a topic could be advantageous for some people.

5. Conclusion

In this paper, we propose a method based on Latent Dirichlet Allocation and support vector machines for identifying and collecting topic relevant hashtags in Twitter streams. Unlike most proposed methods, our approach relies on a small database only for training the classifier and always searches for the latest tweets. This way, the hashtag network is always as updated as possible. However, it was very rare for the collected hashtags to have more than one degree of distance to the seed hashtag.

We started by searching a small group of tweets from a seed hashtag that represents the topic and stored them in a database. This tweets were later used to generate a topic model and train a SVM to classify new tweets as relevant or irrelevant to the same given topic. Every time we collected a new tweet, we could keep it and harvest its hashtags or discard it based on the classifier. In this manner, we are capable of creating a network of hashtags relevant to the topic.

According to our tests, this method has shown promising results in pruning the network of unrelated hashtags since TORHID was successful in detecting relevant tweets and separating them from the irrelevant ones. One important achievement is the very high precision rate, meaning that once the classifier categorizes a tweet as relevant, that classification is probably accurate.

In addition, we compared TORHID with *Hashtagify*, which is a web service whose goals are similar to ours, with satisfactory results. TORHID not only collected on average around the double of relevant hashtags, as well as a smaller percentage of irrelevant ones, but also provided more information about the relations of those hashtags with each other and the seed, thanks to the network of relations. We were also more successful in collecting hashtags that are relevant depending on the context or time, that is, more updated.

The number of hashtags returned by TORHID was on average fifteen which is very acceptable, specially when compared with the alternatives. There may be some approaches for possibly performing a controlled expansion of the network like retraining the classifier with the new hashtags in the network and redo the search, combining the results of TORHID with other tools like *Hashtagify* or using a community detection algorithm like the Louvain Method in larger networks.

The major drawback with the method we propose is the time required for collecting the training data and for training the models, which takes longer than the alternatives investigated. This compromise is a consequence of keeping the results as updated as possible. However, new approaches will be attempted to achieve faster run times.

Overall, our proposed method to search, expand and prune a network of hashtags related to a topic in Twitter streams provided acceptable results, specially when considering the fact that all stages, including the training of the classifier, are completely unsupervised.

Declaration of Interest

None.

Acknowledgments

This work was conducted as part of the RECAP preterm Project which received funding from the European Unions Horizon 2020 research and innovation program [grant number 733280].

References

[1] D. Antenucci, G. Handy, A.N. Modi, M. Tinkerhess, Classification of tweets via clustering of hashtags, in: EECS 545, Final Project, 2011, pp. 1–11.
[2] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.
[3] G. Csardi, T. Nepusz, The igraph software package for complex network research, InterJournal Complex Systems (2006) 1695.
[4] M. Efron, Hashtag retrieval in a microblogging environment, in: F. Crestani, S. Marchand-Maillet, H. Chen, E. Efthimiadis, J. Savoy (Eds.), Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19–23, 2010, ACM, 2010, pp. 787–788, doi:10.1145/1835449.1835616.
[5] J. Gentry, twitteR, 2016. R package version 1.1.9
[6] S. GmbH, Most famous social network sites 2017, by active users, 2017.
[7] S. GmbH, Twitter: number of monthly active users 2010–2017, 2017.
[8] F. Godin, V. Slavkovikj, W.D. Neve, B. Schrauwen, R.V. de Walle, Using topic models for twitter hashtag recommendation, in: WWW (Companion Volume), International World Wide Web Conferences Steering Committee / ACM, 2013, pp. 593–596.
[9] B. Grün, K. Hornik, Topicmodels: an r package for fitting topic models, Journal of Statistical Software 40 (13) (2011) 1–30, doi:10.18637/jss.v040.i13.
[10] S.M. Kywe, T. Hoang, E. Lim, F. Zhu, On recommending hashtags in twitter networks, in: K. Aberer, A. Flache, W. Jager, L. Liu, J. Tang, C. Guéret (Eds.), Social Informatics - 4th International Conference, SocInfo 2012, Lausanne, Switzerland, December 5–7, 2012. Proceedings, Lecture Notes in Computer Science, 7710, Springer, 2012, pp. 337–350, doi:10.1007/978-3-642-35386-4_25.
[11] O. Laughland, R. Luscombe, A. Yuhas, Florida school shooting: at least 17 people dead on 'horrific, horrific day', The Guardian (2018).
[12] Q. Li, S. Shah, R. Fang, A. Nourbakhsh, X. Liu, Discovering relevant hashtags for health concepts: A case study of twitter, in: AAAI Workshop: WWW and Population Health Intelligence, in: AAAI Workshops, WS-16-15, AAAI Press, 2016, pp. 783–786.
[13] Z. Li, D. Zhou, Y. Juan, J. Han, Keyword extraction for social snippets, in: M. Rappa, P. Jones, J. Freire, S. Chakrabarti (Eds.), Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26–30, 2010, ACM, 2010, pp. 1143–1144, doi:10.1145/1772690.1772845.
[14] A. Mazzia, J. Juett, Suggesting hashtags on twitter, EECS 545 Project (2010).
[15] R. Mehrotra, S. Sanner, W.L. Buntine, L. Xie, Improving LDA topic models for microblogs via tweet pooling and automatic labeling, in: G.J.F. Jones, P. Sheridan, D. Kelly, M. de Rijke, T. Sakai (Eds.), The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28, - August 01, 2013, ACM, 2013, pp. 889–892, doi:10.1145/2484028.2484166.
[16] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, F. Leisch, C.-C. Chang, C.-C. Lin, e1071, 2017. R package version 1.6–8
[17] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: C. Burges, L. Bottou, Z. Ghahramani, K. Weinberger (Eds.), Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States., 2013, pp. 3111–3119.
[18] E. Otsuka, S. Wallace, D. Chiu, Design and evaluation of a twitter hashtag recommendation system, in: B. Desai, A.M. de Almeida, J. Bernardino, E.F. Gomes (Eds.), 18th International Database Engineering & Applications Symposium, IDEAS 2014, Porto, Portugal, July 7–9, 2014, ACM, 2014, pp. 330–333, doi:10.1145/2628194.2628238.
[19] D. Ramage, S.T. Dumais, D.J. Liebling, Characterizing microblogs with topic models, in: W.W. Cohen, S. Gosling (Eds.), Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23–26, 2010, The AAAI Press, 2010, pp. 130–137.
[20] J. She, L. Chen, TOMOHA: topic model-based hashtag recommendation on twitter, in: C. Chung, A. Broder, K. Shim, T. Suel (Eds.), 23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7–11, 2014, Companion Volume, ACM, 2014, pp. 371–372, doi:10.1145/2567948.2577292.
[21] B. Shi, G. Ifrim, N. Hurley, Learning-to-rank for real-time high-precision hashtag recommendation for streaming news, in: J. Bourdeau, J. Hendler, R. Nkambou, I. Horrocks, B. Zhao (Eds.), Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11, - 15, 2016, ACM, 2016, pp. 1191–1202, doi:10.1145/2872427.2882982.
[22] B. Shi, G. Poghosyan, G. Ifrim, N. Hurley, Hashtagger+: efficient high-coverage social tagging of streaming news, IEEE Trans. Knowl. Data Eng. 30 (1) (2018) 43–58, doi:10.1109/TKDE.2017.2754253.
[23] I.L. Stats, Twitter usage statistics, 2018.
[24] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, Berlin, Heidelberg, 1995.
[25] F. Xiao, T. Noro, T. Tokuda, News-topic oriented hashtag recommendation in twitter based on characteristic co-occurrence word detection, in: M. Brambilla, T. Tokuda, R. Tolksdorf (Eds.), Web Engineering - 12th International Conference, ICWE 2012, Berlin, Germany, July 23–27, 2012. Proceedings, Lecture Notes in Computer Science, 7387, Springer, 2012, pp. 16–30, doi:10.1007/978-3-642-31753-8_2.
[26] W. Yih, J. Goodman, V. Carvalho, Finding advertising keywords on web pages, in: L. Carr, D.D. Roure, A. Iyengar, C. Goble, M. Dahlin (Eds.), Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23–26, 2006, ACM, 2006, pp. 213–222, doi:10.1145/1135777.1135813.
[27] E. Zangerle, W. Gassler, G. Specht, Recommending #-tags in twitter, in: Proceedings of the Workshop on Semantic Adaptive Social Web, 2011, pp. 67–78.
[28] W.X. Zhao, J. Jiang, J. Weng, J. He, E. Lim, H. Yan, X. Li, Comparing twitter and traditional media using topic models, in: P.D. Clough, C. Foley, C. Gurrin, G.J.F. Jones, W. Kraaij, H. Lee, V. Murdock (Eds.), Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18–21, 2011. Proceedings, Lecture Notes in Computer Science, 6611, Springer, 2011, pp. 338–349, doi:10.1007/978-3-642-20161-5_34.