Learning Similarity Functions for Topic Detection in Online Reputation Monitoring

Damiano Spina
UNED NLP & IR Group
C/ Juan del Rosal, 16
28040 Madrid, Spain
damiano@lsi.uned.es

Julio Gonzalo
UNED NLP & IR Group
C/ Juan del Rosal, 16
28040 Madrid, Spain
julio@lsi.uned.es

Enrique Amigó
UNED NLP & IR Group
C/ Juan del Rosal, 16
28040 Madrid, Spain
enrique@lsi.uned.es

ABSTRACT

Reputation management experts have to monitor—among others—Twitter constantly and decide, at any given time, what is being said about the entity of interest (a company, organization, personality...). Solving this reputation monitoring problem automatically as a topic detection task is both essential—manual processing of data is either costly or prohibitive—and challenging—topics of interest for reputation monitoring are usually fine-grained and suffer from data sparsity.

We focus on a solution for the problem that (i) learns a pairwise tweet similarity function from previously annotated data, using all kinds of content-based and Twitter-based features; (ii) applies a clustering algorithm on the previously learned similarity function. Our experiments indicate that (i) Twitter signals can be used to improve the topic detection process with respect to using content signals only; (ii) learning a similarity function is a flexible and efficient way of introducing supervision in the topic detection clustering process. The performance of our best system is substantially better than state-of-the-art approaches and gets close to the inter-annotator agreement rate. A detailed qualitative inspection of the data further reveals two types of topics detected by reputation experts: reputation alerts / issues (which usually spike in time) and organizational topics (which are usually stable across time).

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Experimentation, Measurement

Keywords

Online Reputation Monitoring, Similarity Functions, Topic Detection, Twitter

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609621.

1. INTRODUCTION

What are people saying about a given entity (company, brand, organization, personality, etc.) right now? Is there any issue that may damage the reputation of the entity? If so, what actions should be taken about it?

In order to answer such questions for a given client (the entity of interest), reputation experts have to daily monitor Twitter (among others) and discover, at any given time, what is being said about the client. Solving this reputation monitoring problem automatically as an (entity-specific) topic detection task is both essential and challenging [15, 19]. Essential, because real-time online opinions and comments are now key to understand the reputation of organizations and individuals and manage their public relations, and because manual processing of entity-related Twitter streams is very costly and sometimes simply unfeasible. And challenging, because topics of interest for reputation monitoring are usually fine-grained and suffer from data sparsity—unless the client is Apple, Barack Obama or similar.

The largest evaluation effort on topic detection for Online Reputation Monitoring on Twitter to date has been RepLab 2013 [2], where the test collection provided manual annotations by reputation experts on 142,527 tweets referring to 61 different entities (companies in the banking and cars domains, universities, and music bands). From the results of the participant systems it is not clear whether the topic detection process could benefit from training data (which is available in the dataset), and there is no clear evidence on whether Twitter-specific data (such as tweet metadata, hashtags, timestamps, etc.) could be effectively used to improve the results of term-based clustering.

Therefore, in this paper we focus on two related research questions:

1. *Can Twitter signals be used to improve entity-specific topic detection?* Given that tweets are very short by nature, and entity-related topics usually small, it is reasonable to think that any extra information—and Twitter offers many potentially useful signals in addition to plain content—could be useful to solve the problem.

2. *Can previously annotated material be used to learn better topic detection models?* Usually, clustering and topic detection algorithms are unsupervised. However, in a daily reputation monitoring task, there is likely to be some amount of recently seen and (at least partially) annotated information about the entity being

monitored. The question, then, is how to profit from such annotations in the topic detection task.

In order to answer these two questions, we have modeled the topic detection problem as a combination of two tasks:

1. The first is learning tweet similarity: we use all types of Twitter signals (tweet terms and concepts, hashtags, named users, timestamp, author, etc.) to learn a supervised classifier that takes two tweets as input and decides if the tweets belong to the same topic or not. Most of our experimentation is focused on this problem.

2. The second is applying a clustering algorithm that uses the confidence of the classifier above as a similarity measure between tweets. For this step we simply use HAC (Hierarchical Agglomerative Clustering), which is the top performer in similar tasks [6].

We detail our approach in Section 2, describe and discuss the result of our experimentation in Section 3, review related work in Section 4, and summarize our main results in Section 5.

2. APPROACH

2.1 Task Definition

Given an entity (e.g., *Yamaha*) and a set of tweets relevant to the entity in a certain time span, the task consists of identifying tweet clusters, where each cluster represents a topic/event/issue/conversation being discussed in the tweets, as it would be identified by reputation management experts.

Note that this is not a classification task, since topics discussed in a given stream of tweets are not known a priori. Furthermore, this is not a standard Topic Detection setting, because in our scenario each of the tweets must be assigned to a topic. From the perspective of reputation management, reputation alerts—issues that may affect the reputation of the client—must be detected early, preferably before they explode, and therefore the number of tweets involved may be small at the time of detection. That makes the task harder than standard topic detection, mainly due to sparsity issues: topics about a given entity in a short time frame are part of the "long tail" of Twitter topics, and some of them are small even in comparison with the size of the entity-specific Twitter stream.

Table 1 illustrates some examples of tweets belonging to the same topics, extracted from the RepLab 2013 dataset (described in detail in Section 3.1) and corresponding to entities *Maroon 5*, *Yamaha*, *Ferrari*, *Bank of America* and *Coldplay*.

2.2 Modeling Similarity as a Classification Task

Probabilistic generative approaches are a popular strategy to handle topic detection tasks, but might be less appropriate to solve this problem because of data sparsity [27]. Instead, we focus on learning similarity measures between tweets that predict whether two given tweets are about the same topic or not. We explore a wide range of similarity signals between tweets (terms, concepts, hashtags, author, timestamp, etc.) and use them as classification features to learn similarity measures. Similarity measures are, in turn, fed into a competitive clustering algorithm in order to detect topics.

Following the methodology proposed in [5] for a different clustering problem, we model the problem as a binary classification task: given a pair of tweets $\langle d_1, d_2 \rangle$, the system must decide whether the tweets belong to the same topic (`true`) or not (`false`). Each pair of tweets is represented as a set of features (for instance, term overlapping between both tweets), which are used to feed a machine learning algorithm that learns a similarity function. Once we have learned to classify tweet pairs, we take the positive classification confidence as a similarity measure, which is used by a Hierarchical Agglomerative Clustering (HAC) algorithm to identify the topics.

We now detail the learning similarity step and the clustering step. Finally, in Section 2.3 we describe the features used to learn the similarity function.

2.2.1 *Learning a Similarity Function*

Our first goal is to find a classification function that takes two tweets as input and decides if the tweets belong to the same topic or not. Once the pairwise binary classification model is built, its confidence is used as pairwise similarity measure. Formally, let d, d' be two tweets in a set $\mathcal{T}$. We want to learn a boolean function

$$G(d, d') : \mathcal{T} \times \mathcal{T} \rightarrow \{\text{true}, \text{false}\} \quad (1)$$

that says if both tweets belong to the same topic or not. We define a list of features $F_{d,d'} = (f_1(d, d'), f_2(d, d') \ldots f_n(d, d')$, where each of the features is an estimation of the overlap between d, d' according to different signals. Then we estimate the similarity between d, d' as the probability that they belong to the same topic given $F_{d,d'}$:

$$sim(d, d') = P\left(G(d, d')|F_{d,d'}\right) \quad (2)$$

For each entity, we compute the confidence score for all the possible pairs of tweets related to it. The resulting similarity matrix is used by the Hierarchical Agglomerative Clustering (HAC) algorithm [24], with single linkage, that has been proven to perform competitively in clustering tasks such as Web People Search [13, 28]. In HAC there is no need to specify the number of clusters a priori: the first step is to create one cluster for each tweet in the similarity matrix, and then compute for each cluster the similarity to all other clusters. If the highest similarity computed is above a predefined threshold, the two clusters are merged together (agglomerated). A similarity threshold is then used as a stop criterion to get a flat clustering solution. As for "single linkage", it refers to the way in which clusters are compared during the clustering process: in single-link clustering, the similarity between two clusters is computed as the similarity of their most similar members (i.e. it focuses on the area where both clusters are closest to each other). A drawback of this method is that clusters may be merged due to single noisy elements being close to each other, but in practice it seems to be the best choice for problems related to ours [23, 28, 5].

2.3 Similarity Signals

In our study we consider a total of 13 features that capture many types of Twitter signals. Features can be divided in four families: *term features*, that take into account similarity between the terms in the tweets; *semantic features*,

Table 1: Examples of annotated tweets in the RepLab 2013 training dataset.

Entity	Id	Tweet	Topic
Maroon 5	d_1	maroon 5 quedará excitado con las mexicanas (? jajaja.	Promotion of Concerts
	d_2	Oigan ! Creo vendrá a México Maroon 5 quien sabe bien? Quiero ir *n*	
Yamaha	d_3	Just saw Valentino Rossi going into Yamaha's hospitality!! Don't get too excited though, he just attending Agostini's 70th birthday do	MotoGP - User Comments
	d_4	Big piece of 2013 puzzle solved then with Jorge Lorenzo signing a new 2-year deal with Yamaha	
Ferrari	d_5	Alonso pierde la carrera por la mala estrategia de Ferrari, adicional al gran trabajo de Hamilton	(F1) Strategies in the Race
	d_6	Siempre igual Alonso hace el maximo, lo da todo, pero es que las estrategias de Ferrari. . . son para morirse. . .	
	d_7	@alo_oficial:"A ver si podemos confirmar la mejoría del coche, es una buena prueba para Ferrari" #A3F1Canada	(F1) GP of Montreal
	d_8	@alo_oficial Qué crack. La que organizas. En Canadá vince la Ferrari. xD	
	d_9	#F1 Fernando Alonso says Montreal will be 'crucial indicator' for Ferrari's title bid.	
	d_{10}	Vídeo - La Scuderia Ferrari (@InsideFerrari) y Martin Brundle (@MBrundleF1) nos traen el previo del GP de Canadá: URL #F1	
Bank of America	d_{11}	Cons Prod Strategy Manager at Bank of America (Jacksonville, FL) SAME_URL	Vacancy
	d_{12}	Part Time 20 Hours Bartram Lake Village at Bank of America (Jacksonville, FL) SAME_URL	
	d_{13}	Irony: Bank of America is directly across the street from the Dept of the Treasury. Must make it easy to get those bailouts!	Criticism of BofA Bad Behavior
	d_{14}	In 2010 Bank of America seized three properties that were not under their ownership, 'apparently' due to incorrect addresses.	
Coldplay	d_{15}	and so to mourn the loss of may, a trip to see coldplay is in order. i hope they play that uplifting number the scientist.	Fans go to Concert
	d_{16}	Can't get over how fast this day has come !! @coldplay @USER1 @USER2 @USER3	

that model tweet similarity by mapping tweets to concepts in a knowledge base, and then measuring concept overlap between tweets; *metadata*, which indicate whether the tweets have authors, named users (i.e. twitter users mentioned in the tweets), URLs and hashtags in common; and *time-aware features*, which say how close the creation timestamps are for the tweets being compared.

Term Features.

The most obvious signal to take into account is word similarity. Tweets sharing a high percentage of vocabulary are likely to talk about the same topic and hence, to belong to the same cluster. We experiment with three term features that differ in how the terms are weighted:

- `terms_jaccard`. It computes the Jaccard similarity between the set of (unweighted) terms W in the tweets.

$$f_{\text{terms\_jaccard}}(d, d') = \frac{|W_d \cap W_{d'}|}{|W_d \cup W_{d'}|} \quad (3)$$

- `terms_lin_cf`. Lin's similarity [22] can be seen as a weighted variation of Jaccard:

$$f_{\text{terms\_lin\_cf}}(d, d') = \frac{2 \cdot \sum_{w \in W_d \cap W_{d'}} \log \frac{1}{p(w)}}{\sum_{w \in W_d} \log \frac{1}{p(w)} + \sum_{w \in W_{d'}} \log \frac{1}{p(w)}} \quad (4)$$

  where $p(w) = \frac{cf(w)}{\sum_i cf(w_i)}$ and $cf(w)$ is the term frequency in the collection.

- `terms_lin_tfidf`. Similar to `terms_lin_cf`, this variant uses a tf.idf weighting function meant to capture the specificity of the term with respect to the entity of interest [35]. To compute the tf.idf weight, all tweets related to the entity are treated as a pseudo-document D in the collection C:

$$p(w) = \frac{tf(w, D) \cdot \log \frac{N}{df(w)}}{\sum_i tf(w, D_i)} \quad (5)$$

  where $tf(w, D)$ denotes the term frequency of term w in pseudo-document D; $cf(t)$ denotes the term frequency in the collection C and $df(t)$ denotes the total number of pseudo-documents $D_i \in C$ in which the term t occurs at least once.

Semantic Features.

Intuitively, representing tweets with semantics extracted from a knowledge base can be useful to group tweets that do not have words in common. For instance, the tweets d_1 and d_2 about *Maroon 5* in Table 1 can be clustered together because the phrases *mexicanas* and *Mexico* both link to the concept `Mexico`. In some cases this relation could also be captured with stemming, but at the cost of additional false matches. In addition, it might be useful to detect salient terms when word similarity is low. For instance, the Jaccard similarity for tweets d_5 and d_6 is not high, but mapping into Wikipedia matches *Alonso*, *Ferrari* and *estrategia* in both tweets, which lead to a high concept match between them.

In our experiments we adopt an entity linking approach to gather Wikipedia entries that are semantically related to a tweet: the *commonness* probability [26]—based on the intra-Wikipedia hyperlinks—which computes the probability of a concept/entity c being the target of a link with anchor text q in Wikipedia by:

$$commonness(c, q) = \frac{|L_{q,c}|}{\sum_{c'} |L_{q,c'}|} \quad (6)$$

where $L_{q,c}$ denotes the set of all links with anchor text q and target c.

As the dataset contains tweets in two languages, we use both (Spanish and English) Wikipedia dumps. Spanish Wikipedia articles are then translated to the corresponding English Wikipedia article by following the inter-lingual links, using the Wikimedia API.[1]

Tweets are then represented as the *bag-of-entities* derived from linking each n-gram in the content of the tweet to the most probable Wikipedia entity. In case of n-gram overlap, only the longest is considered.

Analogously to term features, we compute the semantic features `semantic_jaccard`, `semantic_lin_cf` and `semantic_lin_tfidf` over the bag-of-entities tweet representation. The feature `semantic_jaccard` is similarly defined by the Best RepLab system [34], detailed in §3.5.

Metadata Features.

- `author`. Two tweets by the same author are more likely to be about the same topic. In Table 1, an example is tweets d_3 and d_4, which are both published by the same MotoGP follower.
- `namedusers`. The number of mentions of named users that co-occur in a pair of tweets also increases the probability that they are about the same issue. See for instance tweets d_7 and d_8, which are both replies to a user (@alo_oficial) which is central to the topic. Another common example are mentions to the official Twitter account of the entity of interest (@ford, @kia, @audi, @shakira, etc.).
- `urls`. Number of URLs that co-occur in a pair of tweets. Tweets that belong to the same cluster may not have high word similarity but might refer to the same URL (for example, d_{11} and d_{12}). This is usually an indication of a topical relationship.
- `hashtags`. Often, hashtags denote topical co-occurrence, and it can be useful to measure their overlap separately (and in addition to) term overlap. d_9 and d_{10} are an example of two topically related tweets that share a hashtag (#F1).

Time-aware Features.

Frequently, topics reflect an ongoing event (such as a live performance of a music group) or conversation. For this reason, close timestamps increase the probability of two tweets being related. For instance, tweets d_{15} and d_{16} were both published in the hour preceding a concert by Coldplay.

We define the features to estimate temporal relation between tweets, given the timestamps t and t', as:

$$f_{\text{time}}(t, t') = \frac{1}{1 + |t - t'|} \tag{7}$$

which takes values between 0 and 1. We turn this equation into three different features, depending on how we represent time: in milliseconds (`time_millis`), hours (`time_hours`) or days (`time_days`).

Note that the author and the timestamp of the tweets are also considered by the Temporal Twitter-LDA system [34], described in §3.5.

[1] http://www.mediawiki.org/wiki/API:Properties

3. EXPERIMENTS

We first describe the dataset used for our experiments and then we analyze the results that help to answer our research questions: first, we study the impact of the different signals in the process of learning a similarity function in §3.2; then, we study the effect of embedding the similarity functions in the Clustering process to solve the Topic Detection task: in §3.3, we investigate the benefits of Twitter-related signals; in §3.4, whether the learning process is effective, and in §3.5 we compare our results with state-of-the-art results on the same corpus, i.e., the best RepLab 2013 systems. Finally, we report the results of a failure analysis that gives some insights into how reputation experts annotate and which the main challenges for automatic systems are.[2]

3.1 Dataset: RepLab 2013

To address our research questions we use the largest Twitter collection for reputation monitoring known to us, the RepLab2013 [2] dataset. The dataset comprises a total of 142,527 manually annotated tweets in two languages: English and Spanish. This set is divided into 61 subsets corresponding to tweets mentioning one of 61 entities belonging to four domains: automotive, banking, universities and music. For every entity, 750 (1,500) tweets were used as training (test) set on average, with a difference of up to six months between tweets in the training test and tweets in the test set.[3] Crawling was performed from June 1, 2012 to December 31, 2012 using each entity's canonical name as query (e.g., "stanford" for Stanford University). Since entity names are often ambiguous, tweets were first annotated with relevance (*Is the tweet about the entity of interest?*) and only relevant tweets were then manually grouped in topics. In our experiments we use the subset of relevant tweets.

In order to better understand the real impact of similarity functions, we have removed from the collection those tweets annotated in the collection as "near-duplicates" (i.e., sharing most terms), which represent 5% of the collection. In Twitter, near duplicates are usually retweets (copies of the original tweet, possibly with some minor addition or change) or the result of posting some online content on Twitter (the user clicks the "post in Twitter" button that most online media offer). Virtually, every topic detection strategy will cluster those near-duplicates together, and that makes more difficult to estimate the real differences between systems.

Our final dataset comprises a total of 100,869 tweets annotated with 8,765 different topics. On average, this corresponds to 544 (1,109) tweets and 57 (87) topics for training (testing) per entity.

Before computing the features, tweets were normalized by removing punctuation, lowercasing, tokenizing by whitespaces and removing stopwords and words with less than three characters.

3.2 Learning Tweet Similarity

Before tackling the topic detection task, we analyze the effectiveness of different signals to learn a similarity function. Given the small size of a tweet, our hypothesis is that

[2] Code and proposed system outputs for the RepLab 2013 Topic Detection Task are publicly available at http://damiano.github.io/learning-similarity-functions-ORM/

[3] Note that training and test are different and disjoint sets for every entity in the collection.

Twitter-specific signals should help building better similarity functions.

We start by building a pairwise classification model using linear kernel SVM[4] [20] We randomly sample 80,000 pairs of tweets from the RepLab 2013 training dataset, keeping the `true` and `false` classes balanced. We run a 10-fold cross-validation on this sample. Table 2 reports results in terms of averaged accuracy (which is a suitable measure as classes are balanced) for different feature combinations.

We use the Student's t-test to evaluate the significance of observed differences. We denote significant improvements with * and ** ($p < 0.05$ and $p < 0.01$, respectively).

The relative differences seen on SVM cannot be directly extrapolated to any Machine Learning algorithm. Therefore, we also compute Maximal Pairwise Accuracy (maxPWA) [5], which is a theoretical upper bound of the effectiveness of different feature combinations, and computes the performance of an ideal Machine Learning algorithm that, for each classification instance, only listens to the features that give the right information[5].

Remarkably, the Pearson correlation between the accuracy of the linear SVM and the theoretical upper bound maxPWA is 0.93. In other words, whenever a set of features gives useful additional information (as reflected in the theoretical upper bound for any learning algorithm), SVM is able to profit in direct proportion to the amount of new useful signal available. Therefore, differences seen with SVM can be generalized to other algorithms.

An inspection of the results in Table 2 shows that:

- In terms of absolute accuracy scores, the quality of the models is low (between 0.56 and 0.63), given that 0.50 is the performance of a random binary classifier. This indicates that the problem is challenging (see Section 3.6 for a qualitative discussion).
- *Time-aware features are useful.* Time-aware features in isolation only reach 0.56 accuracy. However, when added to content signals (`terms_jaccard`, terms, semantics), they contribute to increase performance, with statistical significance, from 0.60 (content signals only) to 0.61^{**} (content plus time-aware features). Therefore, time features give a moderate but useful signal.
- *Semantic features are useful.* Although terms and our semantic features (links to Wikipedia articles) reach the same accuracy in isolation (0.59), their combination reaches 0.60^{**} (2% relative improvement).
- *Metadata is useful.* Likewise, metadata features (0.60 accuracy) also capture additional information with respect to content only: combining both gives 0.62^{**} accuracy (3% improvement).
- *All features give best performance.* Unsurprisingly, combining all features seems to be the best choice, giving an accuracy of 0.63^{**}, which has a statistically significant difference with respect to using terms (0.59, 6% relative improvement).

[4]We tested other machine learning algorithms like Naïve Bayes and Decision Trees, obtaining lower absolute results but similar relative improvements; hence we report results for SVM only.

[5]Given the quadratic cost of computing maxPWA—$0(n^2)$ for n pairs—we use a balanced sample of 8,000 pairs and report the averaged scores over 10 runs.

In summary, most signals in our study are able to improve the classification process with statistical significance over the use of term-based features only, and their combination gives the best performance. Although the absolute performance of the best learned function seems low (0.63 accuracy), we will see in the following sections that, once the classification confidence is used as similarity measure, it leads to the best topic detection performance reported on the RepLab dataset so far.

We now turn to the experiments on the Topic Detection Task. We first compare the effect of considering different Twitter signals in our similarity function (§3.3), then we study the effect of the learning process (§3.4) with respect to an unsupervised alternative, and finally we compare our results with the state-of-the-art (§3.5).

3.3 Topic Detection: Effect of Twitter Signals

We have seen that a classification model that combines all the features is the most accurate. We now use the positive classification confidence score for a pair of tweets as estimation of the similarity between them, and feed the single-link HAC clustering algorithm with this similarity score to detect the topics in the test set, for each of the 61 entities included in the dataset.

In order to answer one of our initial research questions, *Can Twitter signals be used to improve entity-specific topic detection?*, we compare the results of HAC using two learned similarity functions: a baseline using `terms_jaccard` as signal, and our best function, which uses all features.[6] We report results using the official evaluation measures at the RepLab 2013 Topic Detection Task: Reliability & Sensitivity (R&S) [4] and its balanced F-Measure (harmonic mean), $F_1(R, S)$. Note that, in clustering tasks, R&S are equivalent to the well-known BCubed Precision and Recall measures [3].

Figure 1 shows results as macro-averaged R&S in the RepLab 2013 test dataset. Reliability (y-axis), Sensitivity (x-axis) and $F_1(R, S)$ (dot size and numbers) are plotted. Note that $F_1(R, S)$ is not the harmonic mean of the average R&S, but the average of the harmonic mean for each test case (the 61 entities in the test collection). Each dot in a curve represents the output of the HAC algorithm at different similarity thresholds (in percentiles). A lower similarity threshold gives larger clusters, increasing Sensitivity (BCubed Recall) at the expense of Reliability (BCubed Precision).

If we compare using all features with term similarity only (SVM(all)+HAC versus SVM(term_jaccard)+HAC), Figure 1 shows that they have the same maximal value ($F_1(R, S) = 0.47$), but using all features gives more Reliability at high Sensitivity scores. In order to better quantify the differences between the systems, we report two measures that summarize the difference of both curves in a single score: the Area Under the R&S Curve (AUC) and the Mean Average Reliability (MAR), which is the counterpart of the standard IR evaluation measure MAP (Mean Average Precision) for our curves. Table 3 reports both measures for the two systems. As previously, we denote significant improvements with * and ** ($p < 0.05$ and $p < 0.01$, respectively).

[6]Note that we use the expression "Twitter signals" in a broad sense (signals that go beyond terms in the tweet), and therefore we also consider semantic features which are not, strictly-speaking, Twitter-specific signals.

Table 2: **Learning Similarity Functions: SVM Accuracy and Maximal Pairwise Accuracy theoretical upper bound (maxPWA) for different signal combinations.**

Signal Combination	SVM Acc.	maxPWA
time `{milliseconds, hours, days}`	0.56	0.43
metadata `{author, namedusers, urls, hashtags}`	0.58	0.60
`terms_jaccard`	0.59	0.60
semantics `{sem_jaccard, sem_lin_cf, sem_lin_tfidf}`	0.59	0.70
terms `{terms_jaccard, terms_lin_cf, terms_lin_tfidf}`	0.59	0.78
terms + time	0.61	0.86
terms + semantics	0.60	0.87
terms + semantics + metadata	0.62	0.90
terms + semantics + time	0.61	0.91
all	0.63	0.94

Table 3: Topic Detection: Using all signals versus term co-occurrence, comparison of R&S curves with Area Under the Curve and Mean Average Reliability.

System	AUC	MAR
SVM(terms_jaccard)+HAC	0.40	0.59
SVM(all features)+HAC	0.41	0.61*

In terms of Mean Average Reliability, using all features improves over term co-occurrence with statistical significance (3% relative improvement). In terms of AUC, there is a 2% relative improvement but the difference is not statistically significant. Overall, our results suggest that the use of Twitter signals can improve the topic detection process, although the difference is not dramatic.

3.4 Topic Detection: Effect of the Learning Process

Our second research question was: *Can previously annotated material be used to learn better topic detection models?*. Although many clustering problems are unsupervised in nature, supervision in reputation monitoring makes sense: clients are monitored daily, and what has been seen before is annotated and has an effect on how fresh information is processed. Can we profit from such annotations? The case of the RepLab dataset is challenging, because tweets in the training and test sets are separated by up to six months—depending on the entity—and the issues about an entity can change dramatically in Twitter in a period of six months.

We investigate this question by comparing two approaches that use the same signal (term co-occurrence as measured by the Jaccard formula): an unsupervised system, which uses directly the Jaccard measure between two tweets as similarity measure; and a supervised system, that uses our learned similarity function using the Jaccard measure as the only feature for the classifier. In both cases, we feed the HAC algorithm with each of the similarity measures.

Figure 1 includes both curves (terms_jaccard+HAC and SVM(terms_jaccard)+HAC), and shows that there is a substantial difference between them. The supervised system consistently improves the performance of the unsupervised version regardless of how we set the similarity threshold.

Table 4: Supervised versus Unsupervised Topic Detection.

System	AUC	MAR
terms_jaccard+HAC	0.38	0.57
SVM(terms_jaccard)+HAC	0.40	0.59*

Table 4 compares the supervised and unsupervised approaches in terms of AUC and MAR. The supervised system outperforms its unsupervised counterpart with a 2% relative improvement in terms of MAR, which is statistically significant. The difference in terms of of AUC is larger (5%), but is not statistically significant.

Overall, our results indicate that previous annotations can be used to learn better topic models, although differences are not large in our experimental setting. Probably if the time gap between tweets in the training and test sets were smaller (for instance, days instead of months), the effect of learning would be higher.

3.5 Topic Detection: Comparison with State-of-the-Art

The differences we have detected could be irrelevant or misleading if both our baseline and contrastive systems were below state-of-the-art results. Therefore, we compare our approach with two competitive systems from RepLab 2013:

- **Best RepLab** [34]. The best system in the official RepLab 2013 evaluation campaign [2]. Similar to the feature `semantics_jaccard`, this system represents tweets as a bag of Wikipedia entities. After tweets are *wikified*, tweets with a Jaccard similarity higher than the threshold 0.2 are grouped together.

- **Temporal Twitter-LDA** [34] (T.Twitter-LDA). Inspired on Twitter-LDA [39] and Topics Over Time [36], this topic model takes into account the author and the timestamp distributions, in addition to the word distribution in the tweets. In order to estimate the right number of clusters, they incorporate large amounts of additional (unlabeled) tweets to the target data to be clustered and then apply the topic model. We include this system in the comparison because T.Twitter-LDA is a good representative of generative models as com-

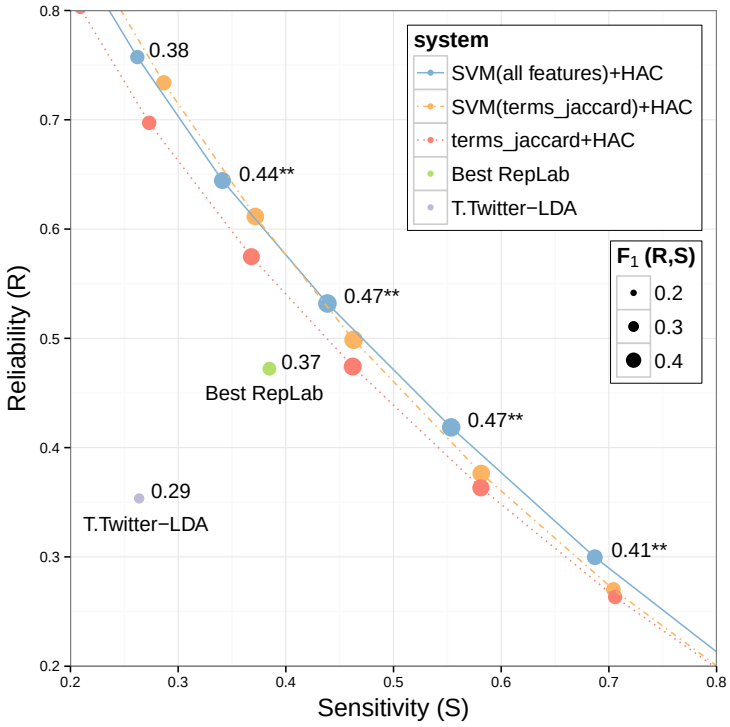

Figure 1: Reliability/Sensitivity curves for the Topic Detection Task. Size of the dot represents $\mathbf{F_1(R,S)}$ averaged over test cases. $\mathbf{F_1(R,S)}$ scores with ** indicate statistically significant improvements with respect to best RepLab system ($\mathbf{p < 0.01}$).

pared to the HAC clustering algorithm that we have used.

Figure 1 compares all the systems. Numbers with ** indicate statistical improvements ($p < 0.01$) of our best system (SVM(all features)+HAC), at different similarity thresholds, with respect to the best RepLab system.[7]

Note that our approach significantly outperforms both the best RepLab system and the T.Twitter-LDA approach, for any reasonable threshold. Note also that a direct application of the HAC algorithm using Jaccard as similarity metric also performs better than the two RepLab systems, which seems to confirm that a standard clustering algorithm may be more robust when there is data sparsity, as is the case of reputation monitoring.

If we compare with inter-annotator agreement, our best system (with $F_1(R,S) = 0.47$) gets very close to the reported annotator agreement on the dataset, which is 0.48, measured as the F_1 score of one annotator vs other [2]. Inter-annotator agreement is low, but this is not surprising for a clustering task, even if annotators are reputation experts. But it may be unrealistic to look for improvements in F_1 beyond what we have reached with our learned similarity measures. It is probably more practical to do failure analysis and study where the challenges of the task lie and what is the performance of our systems on a case-per-case basis. This is what we do in the next section.

[7]Note that R, S and $F_1(R,S)$ for the two RepLab systems reported are different than the official scores [2], because we are excluding unrelated tweets from our evaluation, and we are excluding also near-duplicates (as described in 3.1). Nevertheless, all systems benefit similarly from the normalization and it does not produce any change in the official ranking

3.6 Failure Analysis

So far, we have only investigated average results of our systems across the 61 entities in the RepLab dataset. Here we perform a more detailed analysis of results.

Surprisingly—given the substantial differences between the entities in the dataset—the standard deviation of our best system is low in terms of both R, S and $F_1(R,S)$ (less than 0.09 in all cases). In particular, the F_1 values of our system trained with all features have a standard deviation of 0.06, which compares well with respect to the best RepLab 2013 system (which has a standard deviation of 0.1). Apparently, our system not only performs better on average, but is also more robust across test cases.

In terms of the effect of combining signals, we have seen that taking into account all signals has a slight—but statistically significant—improvement with respect to term matching. If we look case by case, there are only five entities (8% of the whole set) where the average Reliability of using all signals is lower—by a difference of at most 0.02—than using term co-occurrence only: *Capital One*, *Shakira*, *PSY*, *Banco Santander* and *BBVA*. In most cases, for these entities there are large topics that are easy to identify by co-occurrence. For instance, *BBVA* has a topic *Sports sponsor* in which the annotator has grouped all mentions to *BBVA* sports sponsoring activities. The topic covers 52% of the target tweets, and can be identified with a few keywords that have high precision and high recall and refers to the name of the Spanish Soccer League. Likewise, the entity *Shakira* contains a

topic *Charity*, with 92 tweets, that refers to the Barefoot Foundation and can be detected by the keyword *support* or the hashtag *#BuyABrick*.

Finally, we have manually inspected *hard topics*—those where our system either fails to cluster, leaving most tweets in single clusters, or creates just a few big noisy clusters—and *easy topics*—those that are accurately solved by any of the similarity combinations tested in our experiments.

Remarkably, we found that *hard topics* seem to be general, organizational topics that are used by the reputation manager to organize the information in an abstract manner. Some examples are *"Concern of Customers"*, *"Bad Service"*, and *"Hate - Opinions"* for the banking domain, *"Fans Tweeting"* in the music domain or *"Looking Forward to Own a Car"*, *"Negative Opinion of an Owner"* in the automotive domain. In these cases, the content overlap between tweets can be low; for instance, customers complain about the service of their bank in many different ways.

On the other hand, topics easy to find are fine-grained and either refer to specific events—*"Man Arrested for Racial Abuse during Capital One Cup Game"*, *"Cisco Hires Barclays to Sell Linksys"*, *"Barclays Fires or Disciples Staff for LS"*, *"Calls to Condemn Uganda's Politics"*, *"Qatar Selling Warrants"*, *"Dave Matthews Band at Wells Fargo Center"*—or talk about a singular dimension of the entity—*"Lexus Owners Club"*, *"Stock Analysis"*, *"Exchange Rates"*. In general, the vocabulary used in event-like topics tends to be more specific than in organizational topics such as *"Ironic Comments of Costumers"*, reducing the difficulty to identify topical relations.

The nature of hard and easy topics is, therefore, quite different. From the point of view of reputation monitoring, the second type of topics is probably more relevant, as it is where reputation alerts tend to be. Hard topics, on the other hand, seem more like a way of categorizing tweets that do not belong to any significant trending topic, and they are more likely to be used differently by different annotators; perhaps the inter-annotator agreement in the dataset would be higher if we only look at event-like topics. In any case, it is probably useful to make this distinction explicit both when creating test collections and when reporting results for the task.

4. RELATED WORK

We first overview the related work on topic and event detection in Twitter; then we summarize the application of topic models to this task, and we finish discussing the state-of-the-art of topic detection for Online Reputation Monitoring.

Topic and Event Detection in Twitter. The problem of Topic Detection and Tracking (TDT) in texts has been widely studied as event-based organization of newswire stories [1]. In the last years, topic and event detection in Twitter has also become a very active research area [32, 29]. Co-occurrence bursts and temporal signals have been adopted for detecting topics in both the blogosphere [30, 16, 37] and the Twittersphere [25, 8, 37]. Platakis et al.[30] apply Kleinberg's probabilistic automata method [21] to blogs burst modeling and extracting structure from a text stream. Mathioudakis & Koudas [25] group bursty keywords into related groups based on their co-occurrences and Benhardus and Kalita [8] outlines methodologies for using streaming data, e.g., analysis of tf.idf term weighting and normalized term frequency to identify trending topics. Weng et al. [37] and Chen & Roy [11] detect events by grouping a set of signals (words) with similar patterns of bursts using a modularity-based graph partitioning method. Becker et al.[7] analyzed the effectiveness of combining meta-data information (tags, time, location, etc.) to textual data for clustering of social media documents (e.g., Flickr images) according to previously unknown real-life events. As in our case, their results indicated that meta-data was helpful in their Flickr event detection scenario. Unlike the scenarios tackled in previous work, in our setting (i) we are looking at entity-specific topics, which causes data sparsity and (ii) we need to assign each document / tweet to a topic.

Topic Models. Recently, topics models such as LDA [10] and PLSA [17] have been adapted to the context of Twitter. The general assumption is that each author has a certain distribution of topics, while each tweet is associated only to one topic [38, 31]. Hong & Davison [18] conducted experiments on the quality of topics derived from different aggregation strategies. They concluded that topic models learned from messages posted by the same user may lead to superior performance in classification problems. The common characteristic of all previous work is that there are contexts where there is enough information and redundancy to detect temporal bursts of term frequencies. However, in the ORM scenario, the user is interested in a particular entity, that is, only in a tiny subset of the Twitter stream. Detecting topics of a given entity of interest in Twitter roughly corresponds to the *long tail* in Web search scenarios. This lead to data sparsity, which is a bottle-neck for topic models [27].

State-of-the-Art in Online Reputation Monitoring. To our knowledge, RepLab 2013 [2] is the largest Twitter collection for reputation monitoring, and provides the most reliable test collection for the Topic Detection task. Besides the two systems described in detail in §3.5, RepLab participation included both supervised and unsupervised techniques. On one hand, different clustering techniques such as HAC, VOS clustering [9]—a community detection algorithm—and K-star [33] were used by the participants. The most common similarity functions are cosine [9] and Jaccard similarity [33] over terms. Similar to ours, the term clustering approached presented by UNED [34] uses a supervised learned similarity over Twitter signals (authors, URLs, timestamps and hashtags). However, it computes similarities between terms—in order to detect keywords associated to clusters—rather than between tweets.

On the other hand, LIA [14] and UAMCLYR [33] tackled the Topic Detection task as a multi-class classification problem. LIA [14] used Maximum A Posteriori probability of the most pure headwords of the topics in the training set to assign the tweets in the test set. UAMCLYR [33] used standard multi-class classification techniques, such as Naive Bayes and Sequential Minimal Optimization Support Vector Machines (SMO SVM).

Overall, the results of official RepLab systems were the first set of experiments on the RepLab 2013 dataset. As we have seen in our experiments, a HAC algorithm over term similarity outperforms all the RepLab systems: this is another evidence that corroborates the issue of data sparsity in our Online Reputation Monitoring problem.

Apart from the RepLab Topic Detection Task, Chen et al. [12] have recently studied the problem of discovering hot topics about an organization in Twitter. The problem tackled here is slightly different to our scenario: instead of clustering all the tweets related to an entity of interest, they are only interested in detecting the *hot emerging* topics from an initial clustering generated by cosine similarity. Their ground truth does not include clustering relationships between tweets. Instead of this, they align topics with online news and they manually evaluate the aggregated output of different hot topic detection methods to create the ground truth deciding whether a topic is emerging or not.

5. CONCLUSIONS

Online Reputation Management can be seen as the "long tail" of topic detection in Twitter: except for a few exceptions, the volume of information related to a specific organization/company at a given time is orders of magnitude smaller than Twitter trending topics, and this data sparsity makes the problem much more challenging than analyzing Twitter trends.

In this context, our experimental results indicate that (i) Twitter information (authors, timestamps, etc.) can indeed be used to improve topic detection with respect to the use of textual content only. (ii) It is possible to learn effectively from manually annotated topics, using them to improve the estimation of pairwise tweet similarity. (iii) A conventional clustering algorithm (HAC) using our learned similarity functions performs substantially better than state-of-the-art approaches—including Temporal Twitter-LDA—on the same test collection, and gets close to the inter-annotator agreement rate. Our results seem to confirm that, when data is sparse as in our reputation monitoring scenario, conventional clustering—coupled with an effective similarity function—can be more effective than using generative models such as Temporal Twitter-LDA.

A detailed qualitative analysis of our results has revealed that there is a special type of topics in the manual data which are harder to detect automatically. These are *organizational* topics which, rather than grouping tweets about a specific issue or event, have a more taxonomical or structural nature: for instance, a reputation expert may group together all tweets which are *hate opinions* about a bank. Organizational topics tend to be stable across time, and have a wider vocabulary entropy. In contrast, reputation alerts, which are the key issues from a monitoring perspective (e.g., *director of the bank accused of evading taxes*) tend to be spikes in a certain period of time. Organizational topics are not only the main challenge for topic detection systems, but they may also explain the low inter-annotator agreement rates even when, as in the case of the dataset used in our experiments, manual annotations are performed by trained experts. It would be useful, in future test collections, to make this distinction explicit both when creating test collections and when reporting results for the task.

6. ACKNOWLEDGMENTS

This research was partially supported by the European Community's FP7 Programme under grant agreement nr. 288024 (LiMoSINe), the Spanish Ministry of Education (FPU grant nr. AP2009-0507), the Spanish Ministry of Science and Innovation (Holopedia Project, TIN2010-21128-C02), the UNED (project nr. 2012V/PUNED/0004), the Regional Government of Madrid and the ESF under MA2VICMR (S2009/TIC-1542) and a Google Faculty Research Award (Axiometrics Project).

7. REFERENCES

[1] J. Allan. Introduction to Topic Detection and Tracking. In *Topic Detection and Tracking*, volume 12 of *The Information Retrieval Series*, pages 1–16. 2002.

[2] E. Amigó, J. Carrillo de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martín, E. Meij, M. de Rijke, and D. Spina. Overview of RepLab 2013: Evaluating Online Reputation Monitoring Systems. In *Proceedings of CLEF '13*, 2013.

[3] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009.

[4] E. Amigó, J. Gonzalo, and F. Verdejo. A General Evaluation Measure for Document Organization Tasks. In *Proc. of SIGIR'13*, 2013.

[5] J. Artiles, E. Amigó, and J. Gonzalo. The role of named entities in web people search. In *Proceedings of EMNLP'09*.

[6] J. Artiles, J. Gonzalo, and S. Sekine. WePS 2 Evaluation Campaign: Overview of the Web People Search Clustering Task. In *2nd Web People Search Evaluation Workshop (WePS),WWW'09*, 2009.

[7] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of WSDM'10*, 2010.

[8] J. Benhardus and J. Kalita. Streaming Trend Detection in Twitter. *Int. J. Web Based Communities*, 9(1):122–139, 2013.

[9] J. L. A. Berrocal, C. G. Figuerola, and Á. Zazo Rodríguez. REINA at RepLab2013 Topic Detection Task: Community Detection. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, 2003.

[11] L. Chen and A. Roy. Event Detection from Flickr Data through Wavelet-based Spatial Analysis. In *Proceedings of CIKM'09*, 2009.

[12] Y. Chen, H. Amiri, Z. Li, and T.-S. Chua. Emerging Topic Detection for Organizations from Microblogs. In *Proceedings of SIGIR'13*, 2013.

[13] Y. Chen, S. Y. M. Lee, and C.-R. Huang. PolyUHK: A robust information extraction system for web personal names. In *2nd Web People Search Evaluation Workshop (WePS), WWW'09*, 2009.

[14] J.-V. Cossu, B. Bigot, L. Bonnefoy, M. Morchid, X. Bost, G. Senay, R. Dufour, V. Bouvier, J.-M. Torres-Moreno, and M. El-Beze. LIA@RepLab 2013. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[15] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo. Deriving Marketing Intelligence from Online Discussion. In *Proceedings of KDD'05*, 2005.

[16] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information Diffusion Through Blogspace. In *Proceedings of WWW'04*, 2004.

[17] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of SIGIR'99*, 1999.

[18] L. Hong and B. Davison. Empirical study of topic modeling in Twitter. In *Proceedings of the First Workshop on Social Media Analytics*, 2010.

[19] B. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *JASIST*, 60(11):2169–2188, 2009.

[20] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98*, 1998.

[21] J. Kleinberg. Bursty and Hierarchical Structure in Streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.

[22] D. Lin. An information-theoretic definition of similarity. *Proceedings of ICML'98*, 1998.

[23] G. S. Mann. *Multi-document Statistical Fact Extraction and Fusion*. PhD thesis, 2006.

[24] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[25] M. Mathioudakis and N. Koudas. TwitterMonitor: trend detection over the twitter stream. In *Proceedings of SIGMOD'10*, 2010.

[26] E. Meij, W. Weerkamp, and M. de Rijke. Adding Semantics to Microblog Posts. In *Proceedings of WSDM'12*, 2012.

[27] S. Moghaddam and M. Ester. On the Design of LDA Models for Aspect-based Opinion Mining. In *Proceedings of CIKM'12*, 2012.

[28] R. Nuray-Turan, Z. Chen, D. V. Kalashnikov, and S. Mehrotra. Exploiting Web Querying for Web People Search in WePS2. In *2nd Web People Search Evaluation Workshop (WePS), WWW'09*, 2009.

[29] S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to Twitter. In *Proceedings of NAACL'10*, 2010.

[30] M. Platakis, D. Kotsakos, and D. Gunopulos. Discovering Hot Topics in the Blogosphere. In *Proceedings of EUREKA 2008*, 2008.

[31] D. Ramage, S. Dumais, and D. Liebling. Characterizing Microblogs with Topic Models. In *Proceedings of ICWSM-10*, 2010.

[32] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of WWW'10*, 2010.

[33] C. Sánchez-Sánchez, H. Jiménez-Salazar, and W. A. Luna-Ramírez. UAMCLyR at Replab2013: Monitoring task. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[34] D. Spina, J. Carrillo de Albornoz, T. Martín, E. Amigó, J. Gonzalo, and F. Giner. UNED Online Reputation Monitoring Team at RepLab 2013. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[35] D. Spina, E. Meij, M. de Rijke, A. Oghina, M. T. Bui, and M. Breuss. Identifying Entity Aspects in Microblog Posts. In *Proceedings of SIGIR'12*, pages 1089–1090, 2012.

[36] X. Wang and A. McCallum. Topics over Time: A non-Markov Continuous-time Model of Topical Trends. In *Proceedings of KDD'06*, 2006.

[37] J. Weng, Y. Yao, E. Leonardi, and F. Lee. Event Detection in Twitter. Technical Report HPL-2011-98, HP Laboratories, 2011.

[38] W. X. Zhao, J. Jiang, J. He, Y. Song, P. Achananuparp, E.-P. Lim, and X. Li. Topical keyphrase extraction from Twitter. In *Proceedings of ACL'11*, 2011.

[39] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Proceedings of ECIR'11*, 2011.