Proceedings of the Ninth International AAAI Conference on Web and Social Media

On Analyzing Hashtags in *Twitter*

Paolo Ferragina Francesco Piccinno Roberto Santoro
Dipartimento di Informatica
University of Pisa
{ferragina, piccinno, santoro}@di.unipi.it

Abstract

Hashtags, originally introduced in Twitter, are now becoming the most used way to tag short messages in social networks since this facilitates subsequent search, classification and clustering over those messages. However, extracting information from hashtags is difficult because their composition is not constrained by any (linguistic) rule and they usually appear in short and poorly written messages which are difficult to analyze with classic IR techniques.

In this paper we address two challenging problems regarding the "meaning of hashtags"— namely, hashtag relatedness and hashtag classification — and we provide two main contributions. First we build a novel graph upon hashtags and (Wikipedia) entities drawn from the tweets by means of topic annotators (such as TagME); this graph will allow us to model in an efficacious way not only classic co-occurrences but also semantic relatedness among hashtags and entities, or between entities themselves. Based on this graph, we design algorithms that significantly improve state-of-the-art results upon known publicly available datasets.

The second contribution is the construction and the public release to the research community of two new datasets: the former is a new dataset for hashtag relatedness, the latter is a dataset for hashtag classification that is up to two orders of magnitude larger than the existing ones. These datasets will be used to show the robustness and efficacy of our approaches, showing improvements in F1 up to two-digits in percentage (absolute).

1 Introduction

In the last years social networks have enormously grown in popularity and, as a consequence, the volume of data produced everyday by their users has grown too. It is not a surprise that this vast collection of data attracted the attention of several researchers and companies interested in mining, categorizing or searching relevant pieces of information from this huge, unstoppable and unstructured large-scale social stream. The introduction of *tags* as a democratic and user-driven way to organize content in Twitter, Facebook, Google+ and several other social networks is of course a clear evidence of this need.

In this paper we concentrate on *hashtags*, which have been made popular by Twitter. A *hashtag* is a string of characters preceded by the symbol #; it is used as a way to join public discussions (Huang, Thornton, and Efthimiadis 2010), categorize messages or build communities around a specific topic of interest (Laniado and Mika 2010; Yang et al. 2012; Wang et al. 2011). Acronyms or abbreviations are very often used as hashtags, mainly because of the length constraint imposed on tweets, thus making it difficult for a (human or automatic) reader to understand the meaning of a hashtag by just looking at its character composition[1].

The peculiarities of hashtags, as well as the short and noisy/poor content of tweets, make it challenging to solve IR problems formulated over hashtags because of the difficulties encountered by classic approaches hinging on the Bag-of-Word paradigm. In order to overcome these limitations, we clearly need to dig more into the understanding of tweets and hashtags. Following a recent line of research, we augment the TF-IDF vector space model with new *semantic* dimensions based upon entities drawn from Wikipedia or other knowledge bases (see, among the others, (Suchanek and Weikum 2013; Meij, Balog, and Odijk 2014)). The key idea in those approaches is to identify, in the (possibly short) input text, meaningful sequences of terms (also called *mentions*) and link them to unambiguous *entities* (pages) drawn from Wikipedia. Since those entities occur as nodes in the Wikipedia graph, new and more sophisticate methods have been designed that empower classic approaches and thus allow to achieve better solutions for many well-known problems formulated over short and possibly noisy texts, such as tweet classification (Vitale, Ferragina, and Scaiella 2012; Meij, Weerkamp, and de Rijke 2012), news clustering (Scaiella et al. 2012), query understanding and annotation (Carmel et al. 2014; Bordino et al. 2013; Blanco, Ottaviano, and Meij 2015; Cornolti et al. 2014). This justifies the significant interest in the literature about the design and comparison of those topic annotation systems, which are at the core of many applications (Usbeck et al. 2015).

In this paper we start from these achievements and investigate the use of topic annotators in the *understanding of the meaning of hashtags*. Our first step is the construction of the *Hashtag-Entity Graph*: a weighted labeled

[1] As an example, consider the hashtag #tcot which stands for "Top Conservatives on Twitter".

graph made up of hashtags and entities drawn from a set of tweets, properly connected via weighted edges which take into account hashtag-entity and entity-entity relations. Entities are extracted by using one of the best topic annotators to date, namely TagME (Ferragina and Scaiella 2010; Cornolti, Ferragina, and Ciaramita 2013). The HE-graph models via its edges both the classic co-occurrence feature among hashtags and entities and, more crucially, the semantic relatedness among those entities computed accordingly to the underlying Wikipedia graph. By parsing a tweet collection of 10M messages, which we have crawled in December 2013 using the Twitter API, we constructed an HE-graph of about 530k nodes and 22M edges.

As the next step we argue that the HE-graph offers a powerful and semantically-structured representation of tweets and their occurring hashtags, so we deploy these promising features to design new algorithms that hinge upon the HE-graph and solve at the state-of-the-art two difficult IR problems pertaining to the *understanding of hashtags*: hashtag relatedness and hashtag classification. Our algorithms turn out to be language-independent as long as a topic annotator is available for the target language of the tweets.

The relatedness between two hashtags is the core operation in various applications which concern hashtags, both directly and indirectly via tweets, such as clustering, classification or recommendation, just to mention a few (Yang et al. 2012; Meng et al. 2012; Ozdikis, Senkul, and Oguztuzun 2012). However, detecting relatedness between (hash)tags is a difficult task for various reasons: first of all, any relatedness judgment needs to draw upon a vast common sense and knowledge background regarding the meaning (or the meanings) of a hashtag; furthermore, there is an unavoidable subjective facet due to the value/belief system of the observer (e.g. how much are #money and #happiness related?); finally, some (hash)tags are polysemous and so they could be associated with many meanings (e.g. #apple). The additional key difficulties with hashtags are that Twitter is not strictly a folksonomy (Wang et al. 2011) and hashtags have different roles in Twitter than tags in other social media (see e.g. (Yang et al. 2012)); finally, tweets are very short and noisy, and generally contain very few hashtags (if any). These features ask for new modeling and algorithmic approaches as well as datasets to test them.

In this paper, we are the first ones in the literature to explicitly deal with hashtag relatedness as a problem on its own, presenting and validating various relatedness functions on a new dataset consisting of about 1000 hashtag pairs rated as relevant or not by human judges, which we constructed for this purpose. Our study will involve a number of hashtag relatedness functions designed upon classic lexical (TF-IDF) as well as co-occurrence features, plus more sophisticated features which take into account the "semantic" structure inherent in the HE-graph. Our experiments show that the known approaches, which just consider lexical features and co-occurrences of hashtags, obtain a poor error rate of about 11%; whilst our algorithms deploying the peculiar properties of the HE-graph achieve error rates up to 1%. In order to strengthen our achievements we also evaluated the performance of our relatedness functions in the clustering framework, where a k-medoid approach is used upon these functions to cluster hashtags in 8 known clusters (see our classification dataset below). As a final result, our relatedness functions will show a sharp gain in clustering precision (measured via the Adjusted Rand Index (Hubert and Arabie 1985)) with respect to other known measures.

Building upon these promising results which highlight, as argued above, the power and robustness of the HE-graph in modeling the semantic relatedness of hashtags, we attack a well-known IR problem regarding the classification of hashtags in Twitter (Romero, Meeder, and Kleinberg 2011; Posch et al. 2013). This problem has been recently introduced by (Romero, Meeder, and Kleinberg 2011), who studied how the diffusion pattern of tweets among users characterizes the different categories of the hashtags contained in those tweets. More recently (Posch et al. 2013) advanced the previous result by proposing a SVM classifier based on lexical (i.e., TF-IDF) and pragmatic features, which describe how a hashtag is used over time by a large group of users. These authors showed an F1 ranging between 73%-79% over 7 classes and 64 hashtags; a rather small experimental setup indeed. In our paper we provide two contributions with respect to this problem. First, we improve the lexical classifiers above by proposing a new hashtag classifier that hinges on the HE-graph and deploys the taxonomy of Wikipedia categories in order to achieve a consistent improvement in F1 up to +12% (absolute). Moreover, in order to assess the robustness of our proposal, we construct a much larger dataset consisting of more than 5000 hashtags, properly classified in 8 categories, and show that our classification algorithm improves the one in (Posch et al. 2013) in F1 up to +9% (absolute).

Overall, our paper introduces the following contributions:

- We devise the HE-graph as a powerful, yet simple, representation of hashtags and entities occurring in tweets, by leveraging the annotations produced by TagME (Ferragina and Scaiella 2010), one of the best annotator in this setting. The structure of the HE-graph models *classic/semantic features* given by the co-occurrence of hashtag-entities in tweets as well as the semantic relatedness among those entities computed accordingly to the underlying Wikipedia graph. By parsing a tweet collection of 10M messages, crawled in December 2013, we construct an HE-graph of about 530k nodes and 22M edges.
- We investigate the robustness of the HE-graph in modeling the semantic relatedness among hashtags by building a dataset of about 1000 hashtag pairs rated as relevant or not by human judges. We then study a number of hashtag relatedness functions designed upon classic lexical as well as more sophisticated features we specifically built upon the HE-graph, and show that the known approaches, which consider just lexical features and co-occurrences of hashtags, obtain a poor error rate of about 11%; whilst our novel algorithms achieve error rates up to 1%. We strengthen these achievements by checking the robustness of these relatedness measures over a clustering problem. Again, our relatedness functions show a sharp

gain in clustering precision with respect to other known measures.

- Building upon these promising results, we attack the hashtag classification problem and offer two main contributions: (i) a dataset consisting of more than 5000 hashtags, properly classified in 8 categories, thus resulting up to two-order of magnitudes larger than the one proposed in (Posch et al. 2013); and (ii) a novel classification algorithm which hinges on the HE-graph and deploys the taxonomy of Wikipedia categories in order to achieve a consistent improvement in F1 over (Posch et al. 2013) up to +9% (absolute), and provides a robust performance even for smaller training sets.

The HE-graph and the relatedness and classification datasets are available at http://acube.di.unipi.it.

2 Related work

For the sake of space, we just point out the main differences between our proposals and the related literature.

On the HE-Graph. Constructing and exploiting a graph-based data structure to model and extract useful information from a dataset is a popular idea in Information Retrieval (Baeza-Yates and Tiberi 2007). The recent advent of topic-annotation tools, as commented in the introduction, paved the way to constructing graphs which mix together lexical features and entity-based features together with their semantic relatedness measures (see e.g. (Suchanek and Weikum 2013) and refs therein). The proposals closer to our approach are the following ones.

The *Entity-Query graph* (Bordino et al. 2013) consists of two types of nodes: user queries (issued to a search engine) and entities drawn from Wikipedia. Nodes are connected by three types of edges: query-query (if two queries are contiguous in a user query session); entity-query, which connect each entity to the queries that contain it; entity-entity, if a user who submitted a query related to the first entity is likely to search for the second one. This graph was used to tackle the problem of recommending a small and diverse set of queries that are serendipitous and relevant for a given web page. Apart from the different types of nodes, the other significant difference with our HE-graph resides in the definition of the entity-entity links that, in our case, are based on the relatedness measure by (Witten and Milne 2008).

The *Topics graph* (Scaiella et al. 2012) is a bipartite-like graph, whose nodes are either result-snippets (returned by a search engine) or entities drawn from Wikipedia and occurring in those snippets; edges connect either pairs of entities (based on their relatedness) or they connect snippets with their annotated entities. The authors exploited the spectral properties of this graph to produce an appropriate topical clustering of search-engine results. The entity-entity edges recall the ones available in our HE-graph, but instead of snippets we have hashtag nodes which are linked to entities via new weight functions.

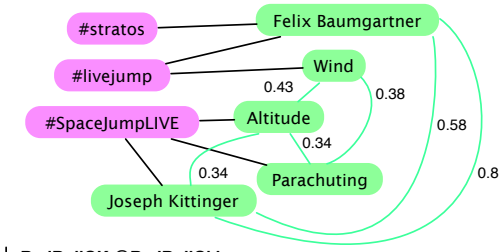
Recently (Sedhai and Sun 2014) introduced an entity-hashtag graph in the context of hashtag recommendation for tweets with hyperlinks. Albeit the name resembles our proposal, our HE-graph is strongly different in structure and goals; the most evident structural difference consists in the definition of "entity" which, in our context, denotes a Wikipedia page that lives in the rich semantic space modeled by Wikipedia's graph (in the spirit of the recent flow of research on entity annotators); moreover the links are not just induced by entity-hashtag co-occurrences but also by entity-entity semantic relations induced by the Wikipedia's graph structure. We will deploy those interconnections to design novel algorithms for "understanding" the meaning of hashtags and thus solve efficiently and efficaciously the two problems addressed in this paper.

On hashtag relatedness. In the literature the terms *relatedness* and *similarity* are often used interchangeably. But, as observed in (Budanitsky and Hirst 2006), semantic relatedness is a more general concept than similarity; similar entities are semantically related by virtue of their similarity (*bank–trust company*), but dissimilar entities may also be semantically related by lexical relationships such as meronymy (*car–wheel*) and antonymy (*hot–cold*), or just by any kind of functional relationship or frequent association (*pencil–paper*, *penguin–Antarctica*, *rain–flood*). Applications typically require relatedness rather than just similarity, therefore in our paper we address the relatedness problem between hashtags.

We will experiment with the most significant known approaches which are either based on co-occurrences of hashtags with non-hashtag words in tweets, context similarity and distributional similarity (Meng et al. 2012; Ozdikis, Senkul, and Oguztuzun 2012), or use some random-walk measures in folksonomies (such as FolkRank (Cattuto et al. 2008)). We will also compare those measures with some newly designed relatedness functions which will be based upon the HE-graph's structural properties.

On hashtag classification. As already described in the introduction, the first work addressing the problem of hashtag classification is due to (Romero, Meeder, and Kleinberg 2011), which prove that is possible to extract the category information of a hashtag by analyzing the diffusion patterns of the tweets that contain it. Then (Posch et al. 2013) proposed the use of a SVM classifier based on both lexical (bag of words) and pragmatic features, the latter ones describing how a hashtag is used over time by a large group of users. We start from this literature and propose a significantly improved classification method for tweets that deploys the HE-graph and the taxonomy of Wikipedia categories, thus achieving better performance than known approaches.

For the sake of completeness we mention (Overell, Sigurbjörnsson, and Van Zwol 2009), who addresses the problem of Flickr tag classification. The authors mapped Flickr tags to Wikipedia pages using anchor texts of Wikipedia, and trained a classifier, which uses Wikipedia Categories and Templates, to classify Wikipedia pages into a set of categories. The idea of using the Wikipedia Category Graph for Wikipedia pages' classification is common to our approach.



RedBullOK @RedBullOkla
"It's all about what we do now and accomplish now.We've made it so far, there's no way we are turning back." - Felix Baumgartner #stratos

The Atlantic @TheAtlantic
Update: Felix Baumgartner's record-breaking #livejump aborted due to wind gusts

Discovery_news @Discovery_News
In 1959, Joe Kittinger set the skydive altitude record; the record has remained for 52 years: #SpaceJumpLIVE

Figure 1: Tweets are at the bottom, the HE-graph is at the top. E_{e-e} is in green (with the weights), E_{h-e} is in black. Notice that the relatedness between `#stratos` and `#SpaceJumpLIVE` might be derived from the relatedness edge between entities `Felix Baumgartner` and `Joseph Kittinger`.

But, unlike this method, the HE-graph provides links from hashtags to entities spotted in several tweets, plus other additional information, such as the annotation weights provided by TagME; all such information can be used to collectively and more robustly infer the topic of a given hashtag.

3 The HE-graph

Our solutions rely upon the HE-graph $G_{HE} = (V_h \cup V_e, E_{h-e} \cup E_{e-e})$, whose structure is derived by parsing a snapshot of Wikipedia and a (possibly large) set $\mathcal{T}$ of tweets. G_{HE} consists of two types of nodes:

- V_h consists of hashtag nodes, one per hashtag occurring in the tweets of $\mathcal{T}$.
- V_e consists of entity nodes, one per entity detected by the annotator TagME in the tweets of $\mathcal{T}$. The choice of TagME is due to its efficient and effective performance over short and poorly composed texts, as tweets are (Cornolti, Ferragina, and Ciaramita 2013). We recall that TagME associates to each entity annotation also a ρ-score which measures the robustness of the resulting annotation. From all the annotations of $\mathcal{T}$ we will discard the ones with $\rho < 0.15$, since they are not reliable (Ferragina and Scaiella 2010).

The HE-graph consists of two types of edges:

- E_{h-e} is the set of directed edges linking a hashtag node h to an entity node e iff they co-occur in a tweet of $\mathcal{T}$. Since h and e can co-occur in more than one tweet, there is one single edge (h, e) labeled with a list containing the ρ-values of the annotations regarding the occurrences of e in $\mathcal{T}$'s tweets; from this list we then derive an edge *weight*, denoted by $w(h, e)$, which counts the cardinality of this list. If $w(h, e) = 1$, the edge is dropped.
- E_{e-e} is the set of undirected edges linking two entities e_1 and e_2 if they are *semantically related*. The semantic relatedness problem has been deeply studied in the literature (see e.g. (Witten and Milne 2008; Ceccarelli et al. 2013) and refs therein); we use the well established measure $rel(e_1, e_2)$ of (Witten and Milne 2008), which does not need any ML-step, it is efficient to be (pre)computed (and stored), and it takes into account the structure of the Wikipedia graph since the more in-pages are shared by the entities/pages e_1 and e_2, the closer their meaning should be, and thus the higher should be their relatedness too. So we set $w(e_1, e_2) = rel(e_1, e_2)$ and drop the edge if its weight is < 0.5, in order to focus only on strong "semantic" entity-entity links.

A concrete example of a HE-graph is provided in Figure 1, where the $h - e$ links capture the co-occurrence between hashtags and entities, whereas the novel semantic dimension is given by the $e - e$ links. Our algorithms will exploit all those features to efficiently and efficaciously solve the two hashtag-understanding problems at hand.

Let us introduce some further useful notation:

- Given a hashtag h, we define LE_h as the set of entities which are linked to h in G_{HE} (*LE* stands for *Linked Entities*). According to the definition of the HE-graph, these are the entities that co-occur with h in some tweets and got a good annotation score.
- We define G_E as the subgraph of G_{HE} restricted to the entity nodes V_e and the relatedness links E_{e-e}. We can look at G_E as the catalog of entities present in the HE-graph and their semantic relations.
- Given an entity e, we define RE_e as the set of entities linked to e in G_E (*RE* stands for *Related Entities*).

4 Hashtag Relatedness

We recall that the investigation on the hashtag relatedness problem has a twofold objective: from one hand, we wish to evaluate whether the semantic dimension induced by the Wikipedia's entities in G_{HE} allows the design of algorithms for the "understanding" of hashtags which improve the classic proposals based on lexical (TF-IDF) features and the co-occurrences of hashtags and terms in tweets; from the other hand, we wish to design relatedness functions which are robust and efficiently computable so that they can be used as a basic block into the successful solution of other problems, such as hashtag clustering.

Our goal is to devise a relatedness function that, given two hashtags $h1$ and $h2$, outputs a real value in $[0, 1]$ which is closer to 1 the more the two hashtags are semantically related, and it is closer to 0 the more the two hashtags are semantically far apart. We designed and tested several hashtag relatedness functions but, because of space restrictions, we discuss below only the ones that are the most interesting algorithmically and/or got the best experimental performance (see Section 7.1).[2] In our design we started from classic ideas — such as vector-space model and cosine similarity, or

[2]For example we experimented all approaches reviewed in (Markines et al. 2009) for tag similarities, such as Jaccard and Dice coefficients, but they gave poor results in our hashtag context.

random walks— and then moved to four more sophisticated measures that exploit the structural properties of G_{HE}.

Baseline (CosText). Our baseline is inspired by (Ozdikis, Senkul, and Oguztuzun 2012) and indeed measures the relatedness between $h1$ and $h2$ by constructing two *meta-documents* consisting of tweets in $\mathcal{T}$ that respectively include $h1$ and $h2$, and then computing the cosine similarity between their TF-IDF vectors.

Labeled LDA (CosLLDA). This was described in (Meng et al. 2012), where hashtags in a tweet are considered labels for a document consisting of the remaining words of that tweet. Then a Labeled LDA model is trained to obtain the hashtag-word distribution, say $p(W_h)$, for the hashtag h. Finally, the relatedness between the two hashtags $h1$ and $h2$ is estimated by computing the cosine similarity between the two vectors $p(W_{h1})$ and $p(W_{h2})$. We also experimented with the negative symmetric KL divergence of the two distributions $p(W_{h1})$ and $p(W_{h2})$, but the results were poorer and so will not be reported in the Experimental Section.

Cosine similarity over entities (CosEntity). Our first new proposal hinges on co-occurrences between hashtags and entities in tweets. We introduce the vector W_h, for the hashtag h, that consists of $|V_e|$ components (one per entity in the HE-graph) and whose j-th component is equal to $w(h, e_j)$. Therefore W_h is a vectorized representation of the hashtag h in the entity space, according to the hashtag-entity co-occurrences in the tweet dataset $\mathcal{T}$. We then compute the hashtag relatedness as the cosine similarity between W_{h1} and W_{h2}. The obvious limitation of this measure is that vector components take into account only entity occurrences, without any deployment of their relatedness relations, so that relatedness is caught only if $LE_{h1} \cap LE_{h2} \neq \emptyset$. In our case this is an unjustified constraint since entities are interconnected by *relatedness links* in G_E, which we would like to exploit in order to move to a sort of *semantic* space.

Expanded cosine similarity (ExpCosEntity). This relatedness measure addresses the limitation above by applying the cosine similarity upon *semantically-expanded* vectors. Given a hashtag h and its weighted vector W_h, we compute the "expanded" vector $\overline{W}_h$, which is obtained by spreading the weight of $e_j \in LE_h$ to all its related entities e_k, taking into account the relatedness between them. Formally, we update W_h as follows: $\overline{W}_h[k] = \overline{W}_h[j] \times rel(e_k, e_j)$ for all entities $e_j \in LE_h$ and for all entities $e_k \in RE_{e_j}$. This way, even if two hashtags have possibly distinct co-occurring entities, namely $LE_{h1} \cap LE_{h2} = \emptyset$, this spreading could make the cosine similarity of the expanded vectors non null, provided that entities in LE_{h1} are related to entities in LE_{h2}. This strongly exploits the semantic space modeled by G_E.

Top-k Relatedness (TopkPairs). In order to be less sensitive to *entity outliers*, which could be introduced in the expansion step above, we decided to investigate a third new measure which estimates the relatedness between the hashtags $h1$ and $h2$ by considering the relatedness in a subset of the pairs of entities in LE_{h1} and LE_{h2}. Let $k = \max(|LE_{h1}|, |LE_{h2}|)$ and consider the k pairs $(e, e') \in LE_{h1} \times LE_{h2}$ having the biggest relatedness values. Then $TopkPairs(h1, h2)$ is the average between these top-k relatedness values. This measure can be seen as a faster and empowered variant of *SimRank* (Jeh and Widom 2002; Quattrone et al. 2011) in which we are deploying the powerful semantic dimension provided by the edges in E_{e-e}.

Personalized PageRank Relatedness (CosPPR). Our fourth and last proposal is based on a random walk over the graph G_E. Given a hashtag h we compute the Personalized PageRank vector over a *directed* version of G_E with *prior probabilities* equal to a properly normalized W_h. The subgraph G_E is made directed by replacing each undirected edge $(e_j, e_k) \in E_{e-e}$ with two directed edges whose weight is equal to $rel(e_j, e_k)$ divided by the out-degree of each source node. So the weight of (e_j, e_k) may be different from the weight of (e_k, e_j), because the two source nodes might have different forward stars. Then we create the Personalized PageRank vector PPR_h for the hashtag h, of $|V_e|$ components, which is computed over the directed G_E with teleportation probability 0.15 and executing 10 iterations (which were enough to reach convergence). We notice that the choice of the normalized W_h as teleportation probability forces the random jump to come back more frequently to entities in LE_h, namely the ones related to h. Finally, we define the relatedness function $CosPPR$ between two hashtags by calculating the cosine similarity between their PPR vectors. We remark that this relatedness function exploits not only the connection between entities in G_E, but also their "volume" and strength, as indeed random-walks approaches are suitable to do.

In Section 7.1 we will experiment these relatedness functions, thus testing the efficacy of the three dimensions (lexical, co-occurrence, semantic) suitably encoded in G_{HE}.

5 Hashtag Classification

The problem we address in this section consists in labeling a given hashtag h with a category topic $c \in \mathcal{C}$ (e.g., $\mathcal{C} = \{sports, movies, \cdots\}$). Albeit the methodology is general enough to handle multi-label classification, we deliberately chose to study single-label classification in order to simplify the construction of the classification dataset by our human judges and to be in line with the works of (Romero, Meeder, and Kleinberg 2011; Posch et al. 2013).

Our approach hinges on two main ingredients: the HE-graph G_{HE} *and* the category information available for each entity $e \in V_e$ in the Wikipedia Category graph, a directed graph in which the edge (x, y) indicates that x is a subcategory of y (x is more specific than y).

Our classification algorithm works as follows. Given a hashtag h that we wish to classify, the algorithm performs a BFS traversal of the Wikipedia Category graph for each entity $e \in LE_h$ (i.e., h is linked to e in G_{HE}). The BFS stops after visiting l levels (the value of l will be discussed in Section 7.2) in the Wikipedia Category graph. During the BFS traversals, the algorithm keeps track in an array CV_h of the number of times each Wikipedia category page is encountered. [3] The rationale behind this simple scheme is to

[3] Other scoring mechanisms, such as scaling by the average ρ of

promote categories that are shared by the vast majority of entities in LE_h by giving them a greater score. This way we are focusing on a set of category pages that best describe the hashtag h under classification.

After the BFS traversals of all $e \in LE_h$ have been completed, the counter-array CV_h is rescaled (by dividing each element by $\max(CV_h)$) and given in input to a SVM classifier with linear kernel. The SVM will be trained/tested via proper ground-truth datasets (more details in Section 7.2).

To better understand the algorithm, consider the hashtag `#iphone`, which co-occurs with many entities such as "`Apple Inc`" and "`iPhone`", so that their BFS traversals will pass through the "*Smartphone*" category node many times, thus increasing its score. On the other hand, less related category nodes such as "*New York*" will possibly receive a non-zero score which will be nonetheless low (e.g., some tweets could talk about queues at the Apple store in NY); the score should thus have very little influence on the final decision of the SVM classifier.

This simple scheme mimics the way in which BoW classifiers work, where the basic feature is the presence or frequency of a word in a text. However the Bow-approach suffers from the sparsity of the tweet representation given their short and poor composition, and from the independent role of each term. Conversely, our classification algorithm surpasses these limitations because each entity is connected to many different category pages whose aggregated scores reasonably describe the semantic meaning of those entities. This way the interconnections in the Wikipedia category graph and the entity-annotation performed by TagME can be used to derive a descriptive fingerprint of the semantic meaning of h.

These considerations will be further explored in the following Experimental Section, where we will analyze and compare the performance of a SVM classifier built upon lexical features (BoW), semantic features (Wikipedia Categories) and both of them together (mixed model).

6 Datasets

Posch dataset. This is the dataset described in (Posch et al. 2013): it is organized in three parts (T_0, T_1, T_2), each accounting for about 95k tweets crawled in four weeks starting March 4th (T_0), April 1st (T_1) and April 29th, 2012 (T_2). It is composed of 64 hashtags divided in 7 categories (between brackets their sizes) as follows: Technology (10), Political (10), Sports (10), Movies (10), Idioms (10), Music (8), Games (6). This dataset presents two main issues: it is small and it includes a peculiar category, i.e. "Idioms", whose hashtags are often ambiguous (consider `#iloveit` and `#MusicMonday`) or detectable by simple parsing rules[4]. We stick to experimenting on this dataset because it allows to compare our classification results with the state-of-the-art algorithms introduced in (Posch et al. 2013).

(h, e), were tested but reported worse performance.

[4]In (Romero, Meeder, and Kleinberg 2011) it is stated that an idioms-hashtag represents a conversational theme on Twitter, consisting of a concatenation of at least two common words.

A larger classification dataset. In order to evaluate the robustness of our approach, we constructed a larger dataset which uses eight categories: *food, games, health, tv/movies, music, politics, sport, technology*. The category *idioms* was removed (see issues above), and two other classic categories were added, that is *food* and *health*.

The hashtags in this dataset have been derived by first downloading the timeline of each verified Twitter user (i.e. about 54k timelines), and then by extracting from those tweets the list of hashtags with frequency > 50. We only considered English tweets, identified via the language attribute returned by the Twitter API. The same extraction process was repeated on a collection of tweets coming from the public stream of Twitter, crawled in November 2013. The two lists of hashtags have then been merged into a set containing more than 91k hashtags.

For each hashtag, we independently asked three human judges to label it with the appropriate category (among the 8 above), if pertinent, using as reference the Twitter Search web page and the definitions coming from Tagdef.[5] In order to reduce the workload of our judges and select potentially interesting and statistically significant hashtags, we restricted the analysis to hashtags which in the timeline co-occur with at least 20 distinct entities and occur in at least 100 tweets. Judges were left with 28k hashtags; only 5245 got a *full agreement* among the judges' classification. These hashtags form the new classification dataset. It is about two orders of magnitude larger than Posch's dataset and it is cleaner because the selection process avoided/reduced polysemous hashtags or erroneous classifications, which instead sometimes occur in Posch dataset as we verified by hand.[6] Details about the class sizes follow:

Food	*Games*	*Health*	*Music*
834	242	549	634
Politics	*Sport*	*Technology*	*TV/Movies*
566	609	911	900

Relatedness dataset. Starting from our larger classification dataset, we built an additional dataset that was used to assess the efficacy and robustness of our relatedness functions with respect to the co-occurrence feature. More precisely we wished to investigate the common issue that two hashtags may co-occur in tweets but this does not necessarily imply that they are related. This analysis is important to assess the robustness of our relatedness functions because techniques based only on hashtag co-occurrence fail to detect such situation. Therefore, we designed a dataset consisting of 911 hashtag pairs, organized in four different parts:

	N^o hashtag pairs	Co-Occurring	Related?
D_1	285	Highly	Yes
D_2	268	Lowly	Yes
D_3	119	Frequently	No
D_4	239	Never	No

[5]http://tagdef.com/

[6]E.g. #avatar is classified in *movies*, but it is also used in Twitter to refer to a user graphical representation in games and social networks, while #wow is classified in *games* but most of the time is used as an exclamation.

The pairs were generated by randomly sampling hashtags contained in the classification dataset, taking into account the frequency of occurrences of the pairs in the tweets of $\mathcal{T}$. More specifically, the second class was built by looking at pairs of hashtags co-occurring only once, while the third class was created using pairs of hashtags co-occurring 10-20 times. We followed the considerations in (Budanitsky and Hirst 2006) as guidelines for collecting human judgments of semantic relatedness. Three judges were asked to categorize the hashtag pairs as related, unrelated, or "Skip" (if the judge does not know the meaning of a hashtag or thinks that the relatedness strongly depends on a subjective facet). In the case of a polysemous hashtag, the judges were instructed to choose the most pertinent meaning. Pairs with full agreement were the only ones selected; as a consequence, the final dataset contains only pairs having a clear relatedness characterization.

The HE-graph. A comprehensive HE-graph is needed to run our algorithms. We constructed this large G_{HE} by crawling, via the Twitter search API, 200 tweets for each hashtag in our original dataset of 91k hashtags. This got about 10M tweets which were parsed with TagME to extract the occurring hashtags and the "robustly-annotated" entities, i.e. having ρ-score ≥ 0.15 (as suggested in (Ferragina and Scaiella 2010)). The resulting graph is quite interesting in size and structure, as the Table below points out.

HE-Graph statistics	
N^o hashtags	348 597
N^o entities	176 626
N^o edges in E_{h-e}	3 995 025
N^o edges in E_{e-e}	17 472 182
N^o zero-degree vertices in G_E	17 577
Avg. V_h out-degree	11.46
Avg. V_e in-degree (without E_{e-e})	22.62
Avg. V_e degree restricted to G_E	219.71
Avg. $\lvert\rho_{i,j}\rvert$	6.8
N^o conn. compo. in G_{HE} (undir.)	27
N^o conn. components in G_E	645
Maximal component in G_E	158 992 nodes

This table shows that the HE-graph is well connected, it basically consists of one huge component and very few other small connected components. The average node eccentricity in the maximal component of G_E is about 7 and the diameter is 9: therefore, not many hops are required to reach any other entity node in that component. The degree distribution of nodes is reported in the following pictures. We notice that three out of four figures remind of a power law, but the last one referring to entity nodes in G_E does not.

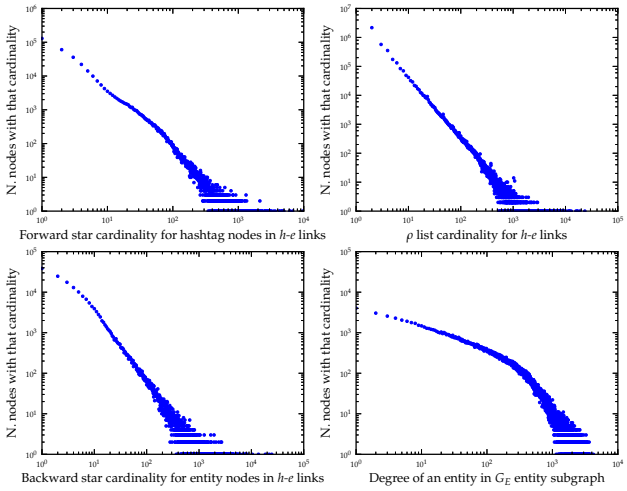


Figure 2: Degree distribution in the HE-graph.

7 Experimental results

7.1 Hashtag relatedness

Before analyzing the results, it is important to point out how the pairs of related hashtags (data-subsets D_1 and D_2) differ with respect to relatedness "strength": we go from pairs representing the same concept (e.g. #doctorwho - #drwho) to pairs expressing a "weaker" relation (e.g. #hansolo - #yoda, #ramen - #tofu). Therefore, even relatively low values (e.g., 0.3) might be reasonable for related hashtags, depending on the pair.

Function	τ	N^o Errors	% Errors	ARI Index
ExpCosEntity	0.31	4	1%	0.8092
TopkPairs	0.62	28	4%	0.6408
CosPPR	0.06	43	6%	0.6490
CosEntity	0.02	73	10%	0.5375
CosText	0.03	82	11%	0.0010
CosLLDA	0.04	86	12%	0.0008

Table 1: Optimal τ values (728 pairs in the test set)

That said, we want to find a threshold τ for each relatedness function such that two hashtags are considered related only if the value returned by the function is $\geq \tau$. We used a small part of the data (20% of each subset) as training set to find the best τ, then tested the relatedness function using the rest of the dataset. Results are reported in Table 1. The best results are obtained by our novel functions which exploit information in G_E. We also point out that $CosLLDA$ does not bring any improvement with respect to our baseline.

CosPPR deserves a special comment because, although sophisticate, it achieves worse results than *ExpCosEntity* and *TopkPairs*. Given W_h, the random walk on G_E boosts the most "semantically" important entities related to h. If h is polysemous or used in a variety of different contexts,[7] then W_h is spread over several topically different entities so that all possible semantic areas for h are correctly taken into account, according to their relative importance given by W_h. Conversely, *TopkPairs* would possibly drop some interesting areas by restricting its computation to the top-k most related pairs, which could come from few areas indeed. This is a very nice property of *CosPPR* which, however, is not fully exploited in this setting. So we envision a system that, given two hashtags, is able to recognize if they are strongly, weakly or not related. We made a test in this respect which

[7] e.g. consider #apple or #montecarlo

has given very promising results by simply considering the outputs of *ExpCosEntity* and *CosPPR*: if they are both high (low), then the two hashtags are related (unrelated); if the output of *CosPPR* is significantly lower than the one of *ExpCosEntity*, then they are weakly related. We plan to further investigate this task in the future.

Function	D_1	D_2	D_3	D_4
ExpCosEntity	0.00%	1.87 %	0.00 %	0.00%
TopkPairs	0.88%	2.80 %	21.05%	0.00%
CosPPR	0.00%	5.61 %	32.63%	0.00%
CosEntity	0.00%	5.14 %	64.21%	0.52%
CosText	0.00%	7.94 %	67.37%	0.52%
CosLLDA	0.00%	12.62%	55.79%	3.14%

Table 2: Error rates with respect to each data-subset

Table 2 details the error rates of each function on each data-subset (of course evaluated on the test subsets). The first thing to point out is that all functions perform well on D_1 and D_4, when related (unrelated) hashtags are highly (not) co-occurrent. Therefore the difference in the overall performance is due to D_2 and D_3, where the co-occurrence is not indicative of relatedness between two hashtags. In particular, the highest error rates are obtained when the hashtags are unrelated but frequently co-occurring, except for *ExpCosEntity* which is undoubtedly the most well-behaving function over all subsets, being robust with respect to the number of co-occurrences.

Clustering. In order to strengthen our evaluation of the four relatedness functions introduced in this paper we set up a clustering experiment, where the aim is to clusterize the hashtags contained in the relatedness dataset into 8 distinct clusters, as many as the category topics of our classification dataset. Suppose we use the relatedness functions as similarity measures between two hashtags that we wish to cluster, and the categories as ground-truth labels. If we cluster the hashtags in eight clusters, we expect that hashtags belonging to the same category will also belong to the same cluster. The objective of this experiment is to determine how good the clusters derived using each relatedness function are with respect to the ground truth category labels. To this end we choose to run the K-Medoids (Kaufman and Rousseeuw 2009) algorithm with $K = 8$ because of its properties: (i) it allows us to specify the number of clusters; (ii) it chooses data points as centers instead of calculating centroids; (iii) it is more robust to noise and outliers with respect to K-Means.

We then calculated the ARI (Adjusted Rand Index) index (Hubert and Arabie 1985) for each relatedness function [8]. The results of this experiment are reported in the last column of Table 1. We recall that ARI values are in the range $[-1.0; 1.0]$, where random labelings have an ARI score close to 0.0 whereas a perfect labeling has a score equal to 1.0. The worst performing relatedness functions are the two baselines, whose ARI score is near to 0. Conversely the functions exploiting the HE-Graph exhibit much better scores. A point worth noting is that functions with higher ARI score are the ones with lower error rate in the previous experiment. Thus the results confirm both the significance of our previous experiment and the applicability of our functions in real-world applications.

[8]Since K-Medoids is a randomized algorithm, the maximum ARI score is reported, out of 100 randomized runs

7.2 Hashtag classification

Our classification algorithm depends only on the parameter l, i.e. the maximum level used during the BFS traversal of the Wikipedia Category graph. After an extensive set of experiments we empirically found that 3 is a good value. Set this value, we experimentally evaluated the performance of our classification algorithm on the two available datasets and got the following results.

Posch dataset. A comparison between our approach and the best classifier of (Posch et al. 2013) is reported in Figure 3 for both time periods T_1 and T_2, using 6-fold cross-validation as in the original paper.[9] Our classifier gets an average F1 which is from +8% up to +12% better (in absolute terms) than the best lexical classifier reported in (Posch et al. 2013). It goes without saying that the small size of this dataset does not allow us to draw conclusions about the robustness of the tested classifiers. The experiment over our larger dataset is meant to establish such robustness.

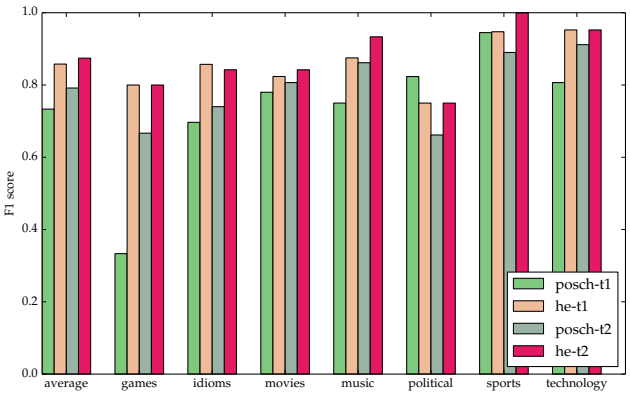


Figure 3: 6-fold cross validation on Posch dataset

Large dataset. We also tested the classifiers on the larger classification dataset we explicitly built for this task. Since we were not able to get the original software of (Posch et al. 2013), we implemented our own version[10] following the details provided in that paper: it consists of a SVM classifier with a linear kernel trained over a standard BoW feature set, i.e. term frequencies (TF).

A detailed comparison between our solution and the lexical classifier is summarized in Table 3. In addition to offering a complete overview of the F1 score for each category topic, the table provides a side-by-side comparison of the performance of each classifier when 5%, 25%, 50% and 75% of the dataset is reserved for training. A graphical but

[9]Figures for (Posch et al. 2013) have been derived from a personal communication with Lisa Posch.

[10]We used sklearn: http://scikit-learn.org/

Name	avg	food	games	health	movies	music	polit.	sports	tech.	N. Errors	Precision	Recall
posch-05%	.852	.951	.636	.857	.852	.902	.859	.923	.836	634	.900	.831
mix-05%	.939	**.970**	**.885**	**.945**	.912	**.966**	.941	.956	.940	281	**.946**	**.936**
he-05%	**.940**	.968	**.885**	.944	**.913**	**.966**	**.942**	**.958**	**.941**	**280**	.945	**.936**
posch-25%	.922	.970	.845	.921	.897	.942	.930	.951	.920	282	.935	.913
mix-25%	**.957**	**.978**	**.944**	**.939**	**.943**	**.975**	.944	.972	**.960**	**162**	**.957**	**.958**
he-25%	.956	**.978**	.942	**.939**	.940	.974	**.945**	**.973**	.959	166	.956	.956
posch-50%	.943	.977	.921	**.934**	.920	.947	.933	.970	.944	143	.948	.939
mix-50%	**.961**	.978	**.967**	.932	.949	.975	.947	**.974**	**.962**	102	**.959**	**.963**
he-50%	**.961**	**.980**	.963	.932	**.952**	**.976**	**.951**	**.974**	.960	**101**	**.959**	**.963**
posch-75%	.951	.972	.931	**.944**	.933	.944	**.972**	.952	.961	62	.956	.947
mix-75%	**.966**	.976	**.975**	.938	**.963**	**.972**	.968	**.970**	**.965**	**45**	**.967**	**.965**
he-75%	.965	**.979**	**.975**	.934	.958	**.972**	.968	**.970**	**.965**	46	.966	.964

Table 3: F1 comparison between semantic features (*he-*), lexical features (*posch-*), semantic + lexical features (*mix-*)

synthetic representation of the same results is depicted in Figure 4, that compares the average F1 scores for each classifier by varying the training set size.

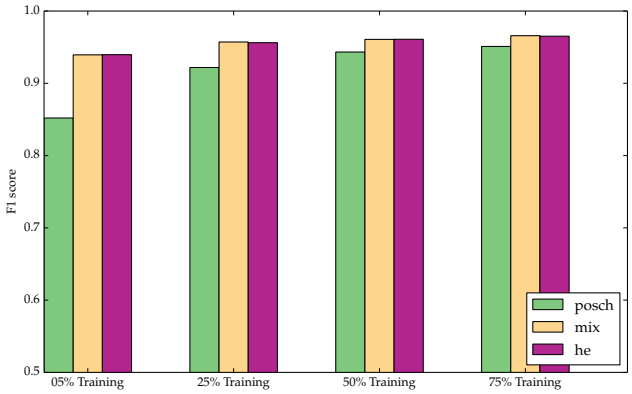


Figure 4: Classification on different training-set sizes

The table and the figure just introduced also compare the performance of a mixed classifier, trained over both lexical (as in (Posch et al. 2013)) and semantic (as in our approach) features. In the table the best performing F1 scores are highlighted using the bold typeface. The two classifiers that use the semantic features have the best performance overall. This is especially true in the case where only a small percentage of the training data (i.e., 5%) is used, where a gain of about 9% in average F1 is obtained by the semantic classifier. The gain starts diminishing when more training data is used, and the three classifiers converge to almost the same F1 score. This is not unexpected at all since the natural sparsity of each tweet is overcome by using a greater amount of data. This does not make our methodology less attractive, but rather shows that our semantic model is better at generalizing with a very little amount of labeled data available, while still being able to obtain a very competitive performance when training data increases. The last three columns of Table 3 respectively report the number of incorrectly classified hashtag, average precision and average recall. As you can see, the performance of the mixed model suggests that the lexical features do not bring any significant improvement to our solution.

The trend is clear: the semantic features derived from Wikipedia provide robustness with respect to training size, a property that is especially desirable in the context of Twitter or other Social Networks, where the amount of information carried in a tweet is very limited and new hashtags are created on a daily basis. This robust performance clearly spurs from the Wikipedia knowledge plugged into both our HE-graph and the Wikipedia Category graph.

8 Conclusions

In this paper we introduced the *Hashtag-Entity Graph* and proper algorithms to efficaciously solve two IR problems formulated on Twitter hashtags: relatedness and classification. We tested our algorithms over known and new datasets, drawn from Twitter, whose size is up to two orders of magnitude larger than the existing ones. These large datasets have been released to the public, together with the HE-graph we constructed. Our experiments systematically show improvements over known approaches. We argue that the HE-graph offers a succinct yet powerful representation for tweets, nicely mixing semantic relatedness between entities and co-occurrence information between hashtags and those entities.

We highlight two important properties of our algorithms: (i) they are language independent (as long as a semantic annotator is available for the target language) and (ii) can be used in a on-line system, in which the users potentially talk about new entities (assuming the knowledge base used behind the scene, like Wikipedia, is updated in near-real-time).

We foresee the application of this graph and the proposed algorithms to other social networks, such as Google+ or Facebook, and to other problems involving hashtags (such as hashtag recommendation) or tweets. In this latter case we argue that the HE-graph could empower the known tools by deploying information derived from the hashtags occurring in those tweets.

9 Acknowledgments

This research has been funded by a Google Research Award and by the Italian MIUR-PRIN grant ArsTechnomedia. The authors warmly thank Lisa Posch for kindly providing the dataset used to assess our hashtag classification algorithm and Massimiliano Ciaramita (Google, Zurich) for fruitful discussions about the topics addressed in this paper. Roberto Santoro warmly thanks SpazioDati, his current employer, for having sponsored his participation in the conference.

References

Baeza-Yates, R., and Tiberi, A. 2007. Extracting semantic relations from query logs. In *KDD*, 76–85. ACM.

Blanco, R.; Ottaviano, G.; and Meij, E. 2015. Fast and space-efficient entity linking in queries. In *WSDM*.

Bordino, I.; De Francisci Morales, G.; Weber, I.; and Bonchi, F. 2013. From machu_picchu to rafting the urubamba river: anticipating information needs via the entity-query graph. In *WSDM*, 275–284. ACM.

Budanitsky, A., and Hirst, G. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics* 32(1):13–47.

Carmel, D.; Chang, M.-W.; Gabrilovich, E.; Hsu, B.-J. P.; and Wang, K. 2014. Erd'14: entity recognition and disambiguation challenge. In *SIGIR*, volume 48, 63–77. ACM.

Cattuto, C.; Benz, D.; Hotho, A.; and Stumme, G. 2008. Semantic analysis of tag similarity measures in collaborative tagging systems. *CoRR 0805.2045*.

Ceccarelli, D.; Lucchese, C.; Orlando, S.; Perego, R.; and Trani, S. 2013. Learning relatedness measures for entity linking. In *CIKM*, 139–148. ACM.

Cornolti, M.; Ferragina, P.; Ciaramita, M.; Schütze, H.; and Rüd, S. 2014. The smaph system for query entity recognition and disambiguation. In *ACM Workshop ERD '14*, 25–30. ACM.

Cornolti, M.; Ferragina, P.; and Ciaramita, M. 2013. A framework for benchmarking entity-annotation systems. In *WWW*, 249–260.

Ferragina, P., and Scaiella, U. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM*, 1625–1628. ACM.

Huang, J.; Thornton, K. M.; and Efthimiadis, E. N. 2010. Conversational tagging in twitter. In *Hypertext*, 173–178. ACM.

Hubert, L., and Arabie, P. 1985. Comparing partitions. *Journal of classification* 193–218.

Jeh, G., and Widom, J. 2002. Simrank: a measure of structural-context similarity. In *KDD*, 538–543. ACM.

Kaufman, L., and Rousseeuw, P. J. 2009. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

Laniado, D., and Mika, P. 2010. Making sense of twitter. In *ISWC*. Springer. 470–485.

Markines, B.; Cattuto, C.; Menczer, F.; Benz, D.; Hotho, A.; and Stumme, G. 2009. Evaluating similarity measures for emergent semantics of social tagging. In *WWW*, 641–650. ACM.

Meij, E.; Balog, K.; and Odijk, D. 2014. Entity linking and retrieval for semantic search. In *WSDM*, 683–684. ACM.

Meij, E.; Weerkamp, W.; and de Rijke, M. 2012. Adding semantics to microblog posts. In *WSDM*, 563–572.

Meng, X.; Wei, F.; Liu, X.; Zhou, M.; Li, S.; and Wang, H. 2012. Entity-centric topic-oriented opinion summarization in twitter. In *KDD*, 379–387. ACM.

Overell, S.; Sigurbjörnsson, B.; and Van Zwol, R. 2009. Classifying tags using open content resources. In *WSDM*, 64–73. ACM.

Ozdikis, O.; Senkul, P.; and Oguztuzun, H. 2012. Semantic expansion of tweet contents for enhanced event detection in twitter. In *ASONAM*, 20–24. IEEE Computer Society.

Posch, L.; Wagner, C.; Singer, P.; and Strohmaier, M. 2013. Meaning as collective use: predicting semantic hashtag categories on twitter. In *WWW*, 621–628.

Quattrone, G.; Capra, L.; De Meo, P.; Ferrara, E.; and Ursino, D. 2011. Effective retrieval of resources in folksonomies using a new tag similarity measure. In *CIKM*, 545–550. ACM.

Romero, D. M.; Meeder, B.; and Kleinberg, J. 2011. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *WWW*, 695–704. ACM.

Scaiella, U.; Ferragina, P.; Marino, A.; and Ciaramita, M. 2012. Topical clustering of search results. In *WSDM*, 223–232. ACM.

Sedhai, S., and Sun, A. 2014. Hashtag recommendation for hyperlinked tweets. In *SIGIR*, 831–834.

Suchanek, F., and Weikum, G. 2013. Knowledge harvesting in the big-data era. In *SIGMOD*, 933–938. ACM.

Usbeck, R.; Röder, M.; Ngonga Ngomo, A.-C.; Baron, C.; Both, A.; Brümmer, M.; Ceccarelli, D.; Cornolti, M.; Cherix, D.; Eickmann, B.; Ferragina, P.; Lemke, C.; Moro, A.; Navigli, R.; Piccinno, F.; Rizzo, G.; Sack, H.; Speck, R.; Troncy, R.; Waitelonis, J.; and Wesemann, L. 2015. GERBIL – general entity annotation benchmark framework. In *WWW (to appear)*.

Vitale, D.; Ferragina, P.; and Scaiella, U. 2012. Classification of short texts by deploying topical annotations. In *ECIR*. Springer. 376–387.

Wang, X.; Wei, F.; Liu, X.; Zhou, M.; and Zhang, M. 2011. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *CIKM*, 1031–1040. ACM.

Witten, I., and Milne, D. 2008. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *AAAI Workshop on Wikipedia and Artificial Intelligence*, 25–30.

Yang, L.; Sun, T.; Zhang, M.; and Mei, Q. 2012. We know what@ you# tag: does the dual role affect hashtag adoption? In *WWW*, 261–270. ACM.