Twinder: A Search Engine for Twitter Streams

Ke Tao, Fabian Abel, Claudia Hauff, and Geert-Jan Houben

Web Information Systems, Delft University of Technology
{k.tao,f.abel,c.hauff,g.j.p.m.houben}@tudelft.nl

Abstract. How can one effectively identify relevant messages in the hundreds of millions of Twitter messages that are posted every day? In this paper, we aim to answer this fundamental research question and introduce Twinder, a scalable search engine for Twitter streams. The Twinder search engine exploits various features to estimate the relevance of Twitter messages (tweets) for a given topic. Among these features are both topic-sensitive features such as measures that compute the semantic relatedness between a tweet and a topic as well as topic-insensitive features which characterize a tweet with respect to its syntactical, semantic, sentiment and contextual properties. In our evaluations, we investigate the impact of the different features on retrieval performance. Our results prove the effectiveness of the Twinder search engine - we show that in particular semantic features yield high precision and recall values of more than 35% and 45% respectively.

1 Introduction

Microblogging sites such as Twitter[1] have emerged as large information sources for exploring and discussing news-related topics [1]. Twitter is also used as a major platform for publishing and disseminating information related to various topics such as politics or sport events[2]. For trending topics, thousands of Twitter messages (tweets) are posted per second. Moreover, the number of posts published per day typically exceeds several hundred million[3]. Thus, searching for tweets that are relevant to a given topic is a non-trivial research challenge.

Teevan et al. revealed that users exhibit a different search behaviour on Twitter compared to Web search [2]. For example, keyword queries on Twitter are significantly shorter than those issued for Web search: on Twitter people typically use 1.64 words to search while on the Web they use, on average, 3.08 words. This can be explained by the length limitation of 140 characters per Twitter message: as long keyword queries easily become too restrictive, people tend to use broader and fewer keywords for searching.

Given the drawbacks of keyword search as provided by Twitter, researchers recently started to investigate alternative search interfaces. Bernstein et al. [3] presented Eddi, an interface which categorizes the tweets in the personal timeline

[1] http://twitter.com/

[2] http://yearinreview.twitter.com/en/tps.html

[3] http://blog.twitter.com/2011/06/200-million-tweets-per-day.html

M. Brambilla, T. Tokuda, and R. Tolksdorf (Eds.): ICWE 2012, LNCS 7387, pp. 153–168, 2012.

of a user into topics and provides access to these tweets by means of tag clouds. In previous work [4], we studied the utility of a faceted search interface for Twitter that allows users to explore topics along different facets such as persons or locations. Moreover, researchers explored various solutions for representing Twitter search results [5], for ranking Web pages that are currently trending on Twitter [6] or for recommending Twitter conversations [7]. However, none of these works focuses on engineering search engines for microblogging data that allow for estimating the relevance of tweets for a given topic.

In this work, we tackle this challenge and introduce *Twinder*, a scalable search engine for Social Web and Twitter streams in particular. Twinder analyzes various features to estimate the relevance of a tweet for a given topic ranging from syntactical characteristics as proposed by [8] (e.g. presence of URLs) to semantic and contextual information (e.g. semantic distance to topic). We explore both *topic-insensitive* features, which can be pre-computed independently from a given topic on a cloud computing infrastructure[4], and *topic-sensitive* features, which are calculated at query time. To analyze the effectiveness of the different features and to investigate how well Twinder can deliver tweets which are *interesting* and *relevant* to a given topic, we evaluated Twinder on a large benchmark dataset of more than 16 million tweets that features relevance judgements for a set of 49 topics. The main contributions of this paper are:

- We present Twinder, a search engine for Twitter streams that analyzes various features in order to identify tweets that are relevant for a given topic. Twinder is designed to run in a cloud computing infrastructure (Section 3).
- We propose methods for extracting novel features from Twitter messages that allow Twinder to predict the relevance of a message for a given topic. In particular, our semantic features go beyond the state of the art (Section 4).
- We evaluate the effectiveness of Twinder for searching Twitter messages and conduct an in-depth analysis to investigate the impact of the different features on retrieval effectiveness (Section 5).

2 Related Work

Since its launch in 2006 Twitter has attracted a lot of attention, both among the general public and among the research community. Researchers started studying microblogging phenomena to find out what kind of topics are discussed on Twitter [1], how trends evolve [9], or how one detects influential users on Twitter [10]. Applications have been researched that utilize microblogging data to enrich traditional news media with information from Twitter [11], to detect and manage emergency situations such as earthquakes [12] or to enhance search and ranking of Web sites which possibly have not been indexed yet by Web search engines.

[4] For example, the supporting website contains a MapReduce-based solution to generate topic-insensitive features [15].

So far though, search on Twitter has not been studied extensively. Tevaan et al. [2] compared the search behaviour on Twitter with traditional Web search behaviour as discussed in the introduction.

Bernstein et al. [3] proposed an interface that allows for exploring tweets by means of tag clouds. However, their interface is targeted towards browsing the tweets that have been published by the people whom a user is following and not for searching the entire Twitter corpus. Jadhav et al. [11] developed an engine that enriches the semantics of Twitter messages and allows for issuing SPARQL queries on Twitter streams. In previous work, we also followed such a semantic enrichment strategy to provide faceted search capabilities on Twitter [4]. Duan et al. [8] investigate features such as Okapi BM25 relevance scores or Twitter specific features (length of a tweet, presence of a hashtag, etc.) in combination with RankSVM to learn a ranking model for tweets. In an empirical study, they found that the length of a tweet and information about the presence of a URL in a tweet are important features to rank relevant tweets.

Our research builds on these previous works. In this paper, we introduce the Twinder search engine for Twitter stream. We re-visit a number of features that were proposed by Duan et al. [8]. Additionally, we also developed a number of novel semantic measures to further boost the retrieval effectiveness of Twinder.

3 Twinder Search Engine

Twinder (*Twi*tter F*inder*) is a search engine for Twitter streams that aims to improve search for Twitter messages by going beyond keyword-based matching. Different types of features ranging from syntactical to contextual features are considered by Twinder in order to predict the relevance of tweets for a given search query. Figure 1 shows the core components of the Twinder architecture. Different components are concerned with extracting features from the incoming messages of a Twitter stream. Given the huge amount of Twitter messages that are published every day, the system is designed to be scalable. For this reason, Twinder makes use of cloud computing infrastructures for processing-intensive tasks such as feature extraction and indexing. Below, we introduce the core components of the *Twinder Search Engine* (see blue boxes in Figure 1) and compare the runtime performances of our engine when running on an cloud computing infrastructure vs. a multi-core server environment.

3.1 Core Components

Feature Extraction. The features used for relevance estimation are extracted by the *Feature Extraction* component. It receives Twitter messages from *Social Web Streams* and implements a suite of functions that allow for representing the tweets via (i) topic-sensitive features which are computed whenever a new query is received and (ii) topic-insensitive features which are calculated when new tweets are received. The set of features that are currently exploited by Twinder are introduced in Section 4. The computation of some features requires

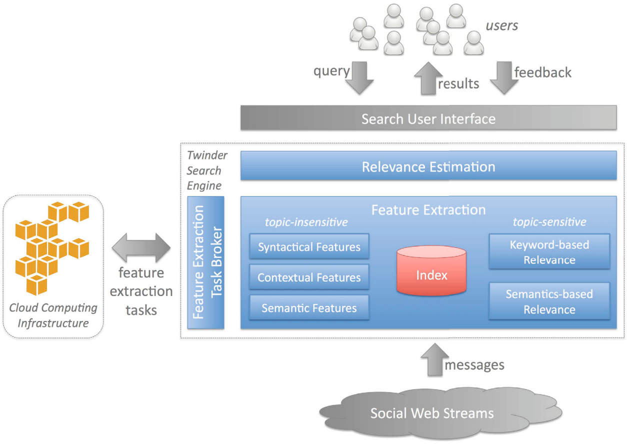

Fig. 1. Architecture of the Twinder Search Engine

further services which offer additional functionalities (e.g. semantic enrichment). Tasks such as obtaining contextual information about the creator of tweets or the actual construction of the multifaceted index of Twinder are repeated periodically. To update its multifaceted index and to compute topic-insensitive features, Twinder provides MapReduce-based implementations and can thus utilize cloud computing infrastructure.

Feature Extraction Task Broker. MapReduce-based implementations are efficient at processing batch tasks with large volume of data and are typically executed on large cloud computing infrastructures. Twinder is designed to take advantage of MapReduce and cloud computing infrastructures to allow for high scalability and to allow for frequent updates of its multifaceted index. For example, the extraction of topic-insensitive features that are not directly available via the Twitter API and the indexing process may be time-consuming tasks for massive amounts of data. Therefore, the *Feature Extraction Task Broker* allows for dispatching feature extraction tasks and indexing tasks to cloud computing infrastructures.

Relevance Estimation. The *Relevance Estimation* component is the most crucial part of Twinder as it determines for a given topic the relevance of tweets which are represented through their calculated features. Technically, the component accepts search queries from a front-end module and passes them to the *Feature Extraction* component in order to compute the features required for the relevance estimation. Tweets that are classified as relevant can be delivered to the front-end for rendering. The relevance estimation is cast into a classification problem where the tweets have to be classified as relevant or non-relevant with respect to a topic. Given a training dataset, Twinder can learn the classification model. At runtime, the learned model is applied to the feature representation of the tweets to identify relevant tweets. In future work, we also

Table 1. Comparison of indexing times: Amazon EMR vs. a single machine

Corpus Size	Mainstream Server	EMR (10 instances)
100k (13MBytes)	0.4 min	5 min
1m (122MBytes)	5 min	8 min
10m (1.3GBytes)	48 min	19 min
32m (3.9GBytes)	283 min	47 min

plan the integration of further user feedback to continuously improve the learned model. For example, we envision the exploitation of re-tweeting activities or favourite markings as additional training data.

3.2 Efficiency of Indexing

As described in Section 3.1, Twinder is capable of leveraging a cloud computing infrastructure to execute data-intensive jobs. In order to demonstrate that it is beneficial to assign tasks with a large amount of data to a cloud computing infrastructure, we compare the performance of creating an inverted index on Amazon ElasticMapReduce (EMR)[5] and a multi-core server[6]. We evaluated the runtime of four different Twitter corpora, ranging in size from 100 thousand to 32 million tweets. On EMR, the indices were built by using ten instances[7], where each instance contains one virtual core, in contrast to the 8 cores in the multi-core server.

As shown in Table 1, if the corpora are small, the index can be efficiently created with a dedicated toolkit on a single machine. However, as the corpus size increases, utilizing cloud infrastructure offers significant speed gains. Therefore we conclude that Twinder can achieve a better runtime performance by employing cloud computing infrastructure.

4 Features of Microposts

In this section, provide an overview of the different features that are analyzed by Twinder.We distinguish two types of features: topic-sensitive features, which are computed at query time and express the relevance of a Twitter message with respect to the query, and topic-insensitive features, which are evaluated before a query is issued and characterize syntactical, semantic or contextual properties of a Twitter message.

4.1 Topic-Sensitive Features

Keyword-Based Relevance Features. A straightforward approach is to interpret Twitter messages as traditional Web documents and apply standard text retrieval measures to estimate the relevance of tweet for a given topic.

[5] `http://aws.amazon.com/elasticmapreduce/`

[6] We wrote our own indexer in Hadoop and relied on the Lemur Toolkit for Information Retrieval to create the index on the single server: `http://www.lemurproject.org/`.

[7] Specifically, we used ten instances of type m1.small.

Feature F1: keyword-based relevance score. To calculate the retrieval score for pair of (topic, tweet), we employ the language modeling approach to information retrieval [13]. A language model θ_t is derived for each document (tweet). Given a query Q with terms $Q = \{q_1, ..., q_n\}$ the document language models are ranked with respect to the probability $P(\theta_t|Q)$, which according to the Bayes theorem can be expressed as follows.

$$P(\theta_t|Q) = \frac{P(Q|\theta_t)P(\theta_t)}{P(Q)} \propto P(\theta_t) \prod_{q_i \in Q} P(q_i|\theta_t). \tag{1}$$

This is the standard query likelihood based language modeling setup which assumes term independence. Usually, the prior probability of a tweet $P(\theta_t)$ is considered to be uniform, that is, each tweet in the corpus is equally likely. The language models are multinomial probability distributions over the terms occurring in the tweets. Since a maximum likelihood estimate of $P(q_i|\theta_t)$ would result in a zero probability of any tweet that misses one or more of the query terms in Q, the estimate is usually smoothed with a background language model, generated over all tweets in the corpus. We employed Dirichlet smoothing [13].

Hypothesis H1: the greater the keyword-based relevance score (that is, the less negative), the more relevant and interesting the tweet is to the topic.

Semantic-Based Relevance Features. Based on the semantics that are extracted from the Twitter messages, we calculate two further relevance features. *Feature F2: semantic-based relevance score.* This feature is a retrieval score calculated as in Section 4.1 though with a different set of terms. Since the average length of search queries submitted to microblog search engines is lower than in traditional Web search, it is necessary to understand the information need behind the query. The search topics provided as part of the TREC data set contain abbreviations, part of names, and nicknames. For example, the name “Jintao” (query: “Jintao visit US”) refers to the president of China. However, in tweets he is also referred to as “President Hu”, “Chinese President”, etc. If these semantic variants of a person’s name are considered when deriving an expanded query then potentially more relevant tweets can be found. We utilize the Named-Entity-Recognition (NER) service DBPedia Spotlight[8] to identify names and their synonyms in the query. We merge the found concepts into an expanded query which is then used as input to the retrieval approach described earlier.

Hypothesis H2: the greater the semantic-based relevance score, the more relevant and interesting the tweet is.

Feature F3: isSemanticallyRelated. It is a boolean value that shows whether there is a semantic overlap between the topic and the tweet. This feature requires us to employ DBpedia Spotlight on the topic as well as the tweets. If there is an overlap in the identified concepts then it is set to *true*.

Hypothesis H3: if a tweet is considered to be semantically related to the query then it is also relevant and interesting for the user.

[8] `http://spotlight.dbpedia.org/`

4.2 Topic-Insensitive Features

Syntactical Features. Syntactical features describe elements that are mentioned in a Twitter message. We analyze the following properties:

Feature F4: hasHashtag. This is a boolean property that indicates whether a given tweet contains a hashtag. Twitter users typically apply hashtags in order to facilitate the retrieval of the tweet. For example, by using a hashtag people can join a discussion on a topic that is represented via that hashtag. Users, who monitor the hashtag, will retrieve all tweets that contain it. Therefore, we investigate whether the occurrence of hashtags (possibly without any obvious relevance to the topic) is an indicator for the interestingness of a tweet.

Hypothesis H4: tweets that contain hashtags are more likely to be relevant than tweets that do not contain hashtags.

Feature F5: hasURL. Dong et al. [6] showed that people often exchange URLs via Twitter so that information about trending URLs can be exploited to improve Web search and particularly the ranking of recently discussed URLs. Hence, the presence of a URL (boolean property) can be an indicator for a relevant tweet.

Hypothesis H5: tweets that contain a URL are more likely to be relevant than tweets that do not contain a URL.

Feature F6: isReply. On Twitter, users can reply to the tweets of other people. This type of communication may be used to comment on a certain message, to answer a question or to chat. For deciding whether a tweet is relevant to a news-related topic, we therefore assume that the boolean *isReply* feature, which indicates whether a tweet is a reply to another tweet, can be a valuable signal.

Hypothesis H6: tweets that are formulated as a reply to another tweet are less likely to be relevant than other tweets.

Feature F7: length. The length of a tweet (the number of characters) may also be an indicator for the relevance. We hypothesize that the length of a Twitter message correlates with the amount of information that is conveyed it.

Hypothesis H7: the longer a tweet, the more likely it is to be relevant.

For the above features, the values of the boolean properties are set to 0 (false) and 1 (true) while the length of a Twitter message is measured by the number of characters divided by 140 which is the maximum length of a Twitter message.

There are further syntactical features that can be explored such as the mentioning of certain character sequences including emoticons, question marks, etc. In line with the *isReply* feature, one could also utilize knowledge about the re-tweet history of a tweet, e.g. a boolean property that indicates whether the tweet is a copy from another tweet or a numeric property that counts the number of users who re-tweeted the message. However, in this paper we are merely interested in original messages that have not been re-tweeted yet and therefore also only in features which do not require knowledge about the history of a tweet. This allows us to estimate the relevance of a message as soon as it is published.

Semantic Features. In addition to the semantic relevance scores described in Section 4.1, we can also analyze the semantics of a Twitter message independently from the topic of interest. We therefore utilize again the NER services provided by DBpedia Spotlight to extract the following features:

Feature F8: #entities. The number of DBpedia entities that are mentioned in a Twitter message may also provide evidence for the potential relevance of a tweet. We assume that the more entities can be extracted from a tweet, the more information it contains and the more valuable it is. For example, in the context of the discussion about birth certificates we find the following two tweets in our dataset:

t_1: *"Despite what her birth certificate says, my lady is actually only 27"*
t_2: *"Hawaii (Democratic) lawmakers want release of Obama's birth certificate"*

When reading the two tweets, without having a particular topic or information need in mind, it seems that t_2 has a higher likelihood to be relevant for some topic for the majority of the Twitter users than t_1 as it conveys more entities that are known to the public and available on DBpedia. In fact, the entity extractor is able to detect one entity, *db:Birth_certificate*, for tweet t_1 while it detects three additional entities for t_2: *db:Hawaii*, *db:Legislator* and *db:Barack_Obama*.

Hypothesis H8: the more entities a tweet mentions, the more likely it is to be relevant and interesting.

Feature F9: diversity. The diversity of semantic concepts mentioned in a Twitter message can be exploited as an indicator for the potential relevance of a tweet. Here, we count the number of distinct types of entities that are mentioned in a Twitter message. For example, for the two tweets t_1 and t_2, the diversity score would be 1 and 4 respectively as for t_1 only one type of entity is detected (*yago:PersonalDocuments*) while for t_2 also instances of *db:Person* (person), *db:Place* (location) and *owl:Thing* (the role *db:Legislator* is not further classified) are detected.

Hypothesis H9: the greater the diversity of concepts mentioned in a tweet, the more likely it is to be interesting and relevant.

Feature F10: sentiment. Naveed et al. [14] showed that tweets which contain negative emoticons are more likely to be re-tweeted than tweets which feature positive emoticons. The sentiment of a tweet may thus impact the perceived relevance of a tweet. Therefore, we classify the semantic polarity of a tweet into positive, negative or neutral using *Twitter Sentiment*[9].

Hypothesis H10: the likelihood of a tweet's relevance is influenced by its sentiment polarity.

Contextual Features. In addition to the aforementioned features, which describe characteristics of the Twitter messages, we also investigate features that

[9] `http://twittersentiment.appspot.com/`

describe the context in which a tweet was published. In our analysis, we focus on the *social context*, which describes the creator of a Twitter message, and investigate the following four contextual features:

Feature F11: #followers. The number of followers can be used to indicate the influence or authority of a user on Twitter. We assume that users who have more followers are more likely to publish relevant and interesting tweets.

Hypothesis H11: the higher the number of followers a creator of a message has, the more likely it is that her tweets are relevant.

Feature F12: #lists. On Twitter, people can use so-called *lists* to group users, e.g. according to the topics about which these users post messages. If a user appears in many Twitter lists then this may indicate that her messages are valuable to a large number of users. Twinder thus analyzes the number of lists in which a user appears in order to infer the value of a user's tweets.

Hypothesis H12: the higher the number of lists in which the creator of a message appears, the more likely it is that her tweets are relevant.

Feature F13: Twitter age. Twitter was launched more than five years ago. Over time, users learn how to take advantage of Twitter and possibly also gain experience in writing interesting tweets. Therefore, we assume that the experienced users are more likely to share interesting tweets with others. Twinder measures the experience of a user by means of the time which passed since the creator of a tweet registered with Twitter.

Hypothesis H13: the older the Twitter account of a user, the more likely it is that her tweets are relevant.

Contextual features may also refer to temporal characteristics such as the creation time of a Twitter message or characteristics of Web pages that are linked from a Twitter message. One could for example categorize the linked Web pages to discover the types of Web sites that usually attract attention on Twitter. We leave the investigation of such additional contextual features for future work.

5 Analysis and Evaluation of Twinder

Having introduced the various features we now turn to analyzing the overall search effectiveness of Twinder. In a second step, we investigate how the different features impact the performance.

5.1 Dataset, Feature Characteristics and Experimental Setup

Dataset. For our evaluations, we use the Twitter corpus which was introduced in the microblog track of TREC 2011[10]. The original corpus consists of approx. 16 million tweets, posted over a period of 2 weeks (Jan. 24 until Feb. 8, inclusive).

[10] The dataset is available via `http://trec.nist.gov/data/tweets/`

Table 2. The dataset characteristics and the relevance prediction across topics. The feature coefficients were determined across all topics. The total number of topics is 49. The five features with the highest absolute coefficients are underlined.

Category	Feature	Relevant	Non-relevant	Coefficient
keyword relevance	keyword-based	-10.699	-14.408	0.1716
semantic relevance	semantic-based	-10.298	-14.206	0.1039
	isSemanticallyRelated	25.3%	4.7%	0.9559
syntactical	hasHashtag	19.1%	19.3%	0.0627
	hasURL	81.9%	53.9%	1.1989
	isReply	3.4%	14.1%	-0.5303
	length (in characters)	90.282	87.819	0.0007
semantics	#entities	2.367	1.882	0.0225
	diversity	1.796	1.597	0.0243
	positive sentiment	2.4%	10.7%	-0.6670
	neutral sentiment	92.7%	82.8%	0.2270
	negative sentiment	4.9%	6.5%	0.4906
contextual	#followers	6501.45	4162.364	0.0000
	#lists	209.119	101.054	0.0001
	Twitter age	2.351	2.207	0.1878

We utilized an existing language detection library[11] to identify English tweets and found that 4,766,901 tweets were classified as English. Employing named entity extraction on the English tweets resulted in a total over 6 million entities among which we found approximately 0.14 million distinct entities. Besides the tweets, 49 search topics were given. TREC assessors judged the relevance of 40,855 topic-tweet pairs which we use as ground truth in our experiments. 2,825 topic-tweet pairs were judged relevant while the majority (37,349) were marked non-relevant.

Feature Characteristics. In Table 2 we list the average values of the numerical features and the percentages of true instances for boolean features that have been extracted by Twinder's feature extraction component. Relevant and non-relevant tweets show, on average, different values for the majority of the features. As expected, the average keyword-based and semantic-based relevance scores of tweets which are judged as relevant to a given topic, are much higher than the ones for non-relevant tweets: -10.7 and -10.3 in comparison to -14.4 and -14.2 respectively (the higher the value the better, see Section 4.1). Similarly, the semantic relatedness is given more often for relevant tweets (25.3%) than for non-relevant tweets (4.7%). For the topic-sensitive features, we thus have first evidence that the hypotheses hold (H1-H3).

With respect to the syntactical features, we observe that 81.9% of the relevant tweets mention a URL in contrast to 53.9% of the non-relevant tweets. Hence, the presence of a URL seems to be a good relevance indicator. Contrary to this, we observe that *hasHashtag* and *length* exhibit, on average, similar values for the relevant and non-relevant tweets. Given an average number of 2.4 entities per tweet, it seems that relevant tweets feature richer semantics than non-relevant

[11] Language detection, `http://code.google.com/p/language-detection/`

Table 3. Performance results of relevance estimations for different sets of features.

Features	Precision	Recall	F-Measure
keyword relevance	0.3036	0.2851	0.2940
semantic relevance	0.3050	0.3294	0.3167
topic-sensitive	0.3135	0.3252	0.3192
topic-insensitive	0.1956	0.0064	0.0123
without semantics	0.3410	0.4618	0.3923
without sentiment	0.3701	0.4466	0.4048
without context	0.3827	0.4714	0.4225
all features	0.3725	0.4572	0.4105

tweets (1.9 entities per tweet). Furthermore, the semantic diversity, i.e. the distinct number of different types of concepts that are mentioned in a tweet, is more than 10% higher for relevant tweets.

As part of the sentiment analysis the majority of the tweets were classified as neutral. Interestingly, Table 2 depicts that for relevant tweets the fraction of negative tweets exceeds the fraction of positive tweets (4.9% versus 2.4%) while for non-relevant tweets it is the opposite (6.5% versus 10.7%). Given the average sentiment scores, we conclude that relevant and interesting tweets seem to be more likely to be neutral or negative than tweets that are considered as non-relevant.

The average scores of the contextual features that merely describe characteristics of the creator of a tweet reveal that the average publisher of a relevant tweet has more followers (*#followers*), is more often contained in Twitter lists (*#lists*) and is slightly older (*Twitter age*, measured in years) than the average publisher of a non-relevant tweet. Given these numbers, we gain further evidence for our hypotheses (H11-H13). Thus, contextual features may indeed be beneficial within the retrieval process.

Experimental Setup. To evaluate the performance of Twinder and to analyze the impact of the different features on the relevance estimation, we relied on logistic regression to classify tweets as relevant or non-relevant to a given topic. Due to the small size of the topic set (49 topics), we use 5-fold cross validation to evaluate the learned classification models. For the final setup of the Twinder engine, all 13 features were used as predictor variables. As the number of relevant tweets is considerably smaller than the number of non-relevant tweets, we employed a cost-sensitive classification setup to prevent the relevance estimation from following a best match strategy where simply all tweets are marked as non-relevant. In our evaluation, we focus on the precision and recall of the relevance classification (the positive class) as we aim to investigate the characteristics that make tweets relevant to a given topic.

5.2 Influence of Features on Relevance Estimation

Table 3 shows the performances of the Twinder's relevance estimation based on different sets of features. Learning the classification model solely based on the keyword-based or semantic-based relevance scoring features leads to an F-measure of 0.29 and 0.32 respectively. Semantics thus yield a better performance than the keyword-based relevance estimation. By combining both types of features (see topic-sensitive in Table 3) the F-measure increases only slightly from 0.3167 to 0.3192. As expected, when solely learning the classification model based on the topic-independent features, i.e. without measuring the relevance to the given topic, the quality of the relevance prediction is extremely poor (F-measure: 0.01). When all features are combined (see *all features* in Table 3), a precision of 0.37 is achieved. That means that more than a third of all tweets, which Twinder classifies as relevant and thus returns as results to the user, are indeed relevant, while the recall level (0.46) implies that our approach discovers nearly half of all relevant tweets. Since microblog messages are very short, a significant number of tweets can be read quickly by a user when presented in response to her search request. In such a setting, we believe such a classification accuracy to be sufficient.

Overall, the semantic features seem to play an important role as they lead to a performance improvement with respect to the F-measure from 0.39 to 0.41. allow for an increase of the F-measure. However, Table 3 also shows that contextual features seem to have a negative impact on the retrieval performance. In fact, the removal of the contextual features leads to a performance improvement in recall, precision and F-measure.

We will now analyze the impact of the different features in more detail.

One of the advantages of the logistic regression model is, that it is easy to determine the most important features of the model by considering the absolute weights assigned to them. For this reason, we have listed the relevant-tweet estimation model coefficients for all involved features in the last column of Table 2. The features influencing the model the most are:

- *hasURL*: Since the feature coefficient is positive, the presence of a URL in a tweet is more indicative of relevance than non-relevance. That means, that hypothesis H5 holds.
- *isSemanticallyRelated*: The overlap between the identified DBpedia concepts in the topics and the identified DBpedia concepts in the tweets is the second most important feature in this model, thus, hypothesis H3 holds.
- *isReply*: This feature, which is *true* (= 1) if a tweet is written in reply to a previously published tweet has a negative coefficient which means that tweets which are replies are less likely to be in the relevant class than tweets which are not replies, confirming hypothesis H6.
- *sentiment*: The coefficient of the positive and negative sentiment features are also strong indicators for estimating the relevance of a tweet which is in line with our hypothesis H8. In particular, the coefficients suggest that negative tweets are more likely to be relevant while positive tweets are more likely to be non-relevant.

We note that the keyword-based similarity, while being positively aligned with relevance, does not belong to the most important features in this model. It is superseded by semantic-based as well as syntactic features. Contextual features do not play an important role in the relevance estimation process.

When we consider the topic-insensitive features only, we observe that interestingness is related to the potential amount of additional information (i.e. the presence of a URL), the overall clarity of the tweet (a reply tweet may only be understandable in the context of the contextual tweets) and the different aspects covered in the tweet (as evident in the diversity feature).

5.3 Influence of Topic Characteristics on Relevance Estimation

In all reported experiments so far, we have considered the entire set of topics available to us. We now investigate to what extent certain topic characteristics impact the performance of Twinder's relevance estimation and to what extent those differences lead to a change in the logistic regression models. Our ambition is to explore to what extent it is useful to adapt Twinder's configuration to the particular type of search topic. We categorized the topics with respect to three dimensions:

- Popular/unpopular: The topics were split into popular (interesting to many users) and unpopular (interesting to few users) topics. An example of a popular topic is *2022 FIFA soccer* (MB002[12]) - in total we found 24. In contrast, topic *NIST computer security* (MB005) was classified as unpopular (as one of 25 topics).
- Global/local: In this split, we considered the interest for the topic across the globe. The already mentioned topic MB002 is of global interest, since soccer is a highly popular sport in many countries, whereas topic *Cuomo budget cuts* (MB019) is mostly of local interest to users living in New York where Andrew Cuomo is the current governor. We found 18 topics to be of global and 31 topics to be of local interest.
- Persistent/occasional: This split is concerned with the interestingness of the topic over time. Some topics persist for a long time, such as MB002 (the FIFA world cup will be played in 2022), whereas other topics are only of short-term interest, e.g. *Keith Olbermann new job* (MB030). We assigned 28 topics to the persistent and 21 topics to the occasional topic partition.

Our discussion of the results focuses on two aspects: (i) the performance differences and (ii) the difference between the models derived for each of the two partitions (denoted $M_{splitName}$). The results for the three binary topic splits are shown in Table 4.

Popularity: We observe that the recall is considerably higher for unpopular (0.53) than for popular topics (0.41). To some extent this can be explained when considering the amount of relevant tweets discovered for both topic splits: while on average 67.3 tweets were found to be relevant for popular topics, only 49.9 tweets were found to be relevant for unpopular topics (the average number of relevant tweets across the entire topic set is 58.44). A comparison of the most

[12] The identifiers of the topics correspond to the ones used in the official TREC dataset.

Table 4. Influence comparison of different features among different topic partitions. There are three splits: popular vs. unpopular topics, global vs. local topics and persistent vs. occasional topics. While the performance measures are based on 5-fold cross-validation, the derived feature weights for the logistic regression model were determined across all topics of a split. For each topic split, the three features with the highest absolute coefficient are underlined.

Performace	Measure	popular	unpopular	global	local	persistent	occasional
	precision	0.3702	0.3696	0.3660	0.3727	0.3450	0.4308
	recall	0.4097	0.5345	0.4375	0.4748	0.4264	0.5293
	F-measure	0.3890	0.4370	0.3986	0.4176	0.3814	0.4750
Category	**Feature**	**popular**	**unpopular**	**global**	**local**	**persistent**	**occasional**
keyword-based	keyword-based	0.1035	0.2465	0.1901	0.1671	0.1542	0.1978
semantic-based	semantic-based	0.1029	0.1359	0.1018	0.0990	0.0808	0.1583
	semantic distance	1.1850	0.5809	0.9853	0.9184	0.8294	1.1303
syntactical	hasHashtag	0.0834	0.0476	0.1135	0.0429	0.0431	0.0803
	hasURL	1.2934	1.1214	1.2059	1.2192	1.2435	1.0813
	isReply	-0.5163	-0.5465	-0.6179	-0.4750	-0.3853	-0.7712
	length	0.0016	-0.0001	0.0003	0.0009	0.0024	-0.0023
semantics	#entities	0.0468	-0.0072	0.0499	0.0107	0.0384	-0.0249
	diversity	-0.0540	0.1179	-0.1224	0.0830	0.0254	0.0714
	negative sentiment	0.8264	0.0418	0.6780	0.3798	0.0707	0.8344
	neutral sentiment	0.2971	0.2102	0.1695	0.2653	0.3723	0.0771
	positive sentiment	-1.0180	-0.3410	-0.7119	-0.6476	-0.6169	-0.6578
contextual	#followers	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	#lists	0.0002	0.0001	0.0002	0.0001	0.0004	0.0001
	Twitter age	0.1278	0.2743	0.0477	0.2646	0.1588	0.2377

important features of $M_{popular}$ and $M_{unpopular}$ shows few differences with the exception of the sentiment features. While sentiment, and in particular positive and negative sentiment, are among the most important features in $M_{popular}$, these features (in particular the negative sentiment) are ranked much lower in $M_{unpopular}$. We hypothesize that unpopular topics do not evoke strong emotions in the users so that sentiment features play a less important role.

Global vs. local: This split did neither result in major differences in the retrieval performances nor in models that are significantly different from each other, indicating that—at least for our currently investigated features—a distinction between global and local topics is not useful.

Temporal persistence: It is interesting to see that the performance (all metrics) is considerably higher for the occasional (short-term) topics than for the persistent (long-term) topics. For topics that have a short lifespan, recall and precision are notably higher than for the other types of topics. In the learnt models, we observe again a change with respect to sentiment features: while the negative sentiment is an important indicator for occasional topics, it is among the least important features for topics that are more persistently discussed on Twitter.

The observation that certain topic splits lead to models that emphasize certain features also offers a natural way forward: if we are able to determine for each topic in advance to which theme or topic characteristic it belongs to, we can select the model that fits the topic best and therefore further optimize the performance of the Twinder search engine.

6 Conclusions

In this paper, we have introduced the Twinder search engine which analyzes various features to determine the relevance and interestingness of Twitter messages for a given topic. We also demonstrated the scalability of the Twinder search engine. In an extensive analysis, we investigated tweet-based and tweet-creator based features along two dimensions: topic-sensitive features and topic-insensitive features. We gained insights into the importance of the different features on the retrieval effectiveness. Our main discoveries about the factors that lead to relevant tweets are as follows:

- The learned models which take advantage of semantics and topic-sensitive features outperform those which do not take the semantics and topic-sensitive features into account.
- Contextual features that characterize the users who are posting the messages have little impact on the relevance estimation.
- The importance of a feature differs depending on the topic characteristics; for example, the sentiment-based features are more important for popular than for unpopular topics.

In the future, we plan to further investigate whether one can adapt the relevance estimation in Twinder to the given search topics. Moreover, we would like to study to what extent personal interests of the users (possibly aggregated from different Social Web platforms) can be utilized as features for personalized retrieval of Twitter messages.

Acknowledgements. The research has received funding from the European Union Seventh Framework Programme, grant agreement no ICT 257831 (ImREAL project).

References

1. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: WWW, pp. 591–600. ACM (2010)
2. Teevan, J., Ramage, D., Morris, M.R.: #TwitterSearch: a comparison of microblog search and web search. In: WSDM, pp. 35–44. ACM (2011)
3. Bernstein, M.S., Suh, B., Hong, L., Chen, J., Kairam, S., Chi, E.H.: Eddi: interactive topic-based browsing of social status streams. In: UIST, pp. 303–312. ACM (2010)
4. Abel, F., Celik, I., Houben, G.-J., Siehndel, P.: Leveraging the Semantics of Tweets for Adaptive Faceted Search on Twitter. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 1–17. Springer, Heidelberg (2011)

5. Golovchinsky, G., Efron, M.: Making sense of twitter search. In: CHI Workshop on Microblogging: What and How Can We Learn From It? (2010)
6. Dong, A., Zhang, R., Kolari, P., Bai, J., Diaz, F., Chang, Y., Zheng, Z., Zha, H.: Time is of the essence: improving recency ranking using twitter data. In: WWW, pp. 331–340. ACM (2010)
7. Chen, J., Nairn, R., Chi, E.H.: Speak Little and Well: Recommending Conversations in Online Social Streams. In: CHI. ACM (2011)
8. Duan, Y., Jiang, L., Qin, T., Zhou, M., Shum, H.Y.: An empirical study on learning to rank of tweets. In: COLING, Association for Computational Linguistics, pp. 295–303 (2010)
9. Mathioudakis, M., Koudas, N.: Twittermonitor: trend detection over the twitter stream. In: SIGMOD, pp. 1155–1158. ACM (2010)
10. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential twitterers. In: WSDM, pp. 261–270. ACM (2010)
11. Jadhav, A., Purohit, H., Kapanipathi, P., Ananthram, P., Ranabahu, A., Nguyen, V., Mendes, P.N., Smith, A.G., Cooney, M., Sheth, A.: Twitris 2.0: Semantically Empowered System for Understanding Perceptions From Social Data. In: Semantic Web Challenge (2010)
12. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: WWW, pp. 851–860. ACM (2010)
13. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to Ad Hoc information retrieval. In: SIGIR, pp. 334–342. ACM (2001)
14. Naveed, N., Gottron, T., Kunegis, J., Alhadi, A.C.: Bad news travel fast: A content-based analysis of interestingness on twitter. In: WebSci. ACM (2011)
15. Tao, K., Abel, F., Hauff, C., Houben, G.J.: Supporting website with additional material (2012), `http://www.wis.ewi.tudelft.nl/twinder/`