Location extraction from tweets

Thi Bich Ngoc Hoang, Josiane Mothe

Location extraction from tweets

Thi Bich Ngoc Hoang[a,b], Josiane Mothe[a,*]

[a] Université de Toulouse and IRIT, UMR5505 CNRS, France
[b] University of Economics, the University of Danang, Vietnam

Keywords:
Information systems
Social networks
Location extraction
Location prediction
Tweets analysis
Predictive model
Machine learning
Microblog collections

A B S T R A C T

Five hundred million tweets are posted daily, making Twitter a major social media platform from which topical information on events can be extracted. These events are represented by three main dimensions: time, location and entity-related information. The focus of this paper is location, which is an essential dimension for geo-spatial applications, either when helping rescue operations during a disaster or when used for contextual recommendations. While the first type of application needs high recall, the second is more precision-oriented. This paper studies the recall/precision trade-off, combining different methods to extract locations. In the context of short posts, applying tools that have been developed for natural language is not sufficient given the nature of tweets which are generally too short to be linguistically correct. Also bearing in mind the high number of posts that need to be handled, we hypothesize that predicting whether a post contains a location or not could make the location extractors more focused and thus more effective. We introduce a model to predict whether a tweet contains a location or not and show that location prediction is a useful pre-processing step for location extraction. We define a number of new tweet features and we conduct an intensive evaluation. Our findings are that (1) combining existing location extraction tools is effective for precision-oriented or recall-oriented results, (2) enriching tweet representation is effective for predicting whether a tweet contains a location or not, (3) words appearing in a geography gazetteer and the occurrence of a preposition just before a proper noun are the two most important features for predicting the occurrence of a location in tweets, and (4) the accuracy of location extraction improves when it is possible to predict that there is a location in a tweet.

1. Introduction

The power of social networking is demonstrated in the vast number of worldwide social network users. According to Statista,[1] this number is expected to reach about 2.5 billion by 2018. Twitter, which enables users to create short, 140 character messages, is one of the leading social networks. The extensive use, speed and coverage of Twitter makes it a major source for detecting new events and gathering social information on events (Weng & Lee, 2011).

As set out in Message Understanding Conference (MUC) campaigns,[2] events have several dimensions that are equally important and require specific attention. The main dimensions are as follows:

- Location information which indicates where the event takes place;

* Corresponding author at: Université de Toulouse and IRIT, UMR5505 CNRS, France.
E-mail addresses: thi-bich-ngoc.hoang@irit.fr (T.B.N. Hoang), josiane.mothe@irit.fr (J. Mothe).

[1] http://www.statista.com/topics/1164/social-networks/.
[2] http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/muc.htm/.

- Temporal information which indicates when the event takes place;
- Entity related information which indicates what the event is about or who its participants are.

This paper focuses on the location dimension. More specifically, it focuses on location extraction from tweets, which is vital to geo spatial applications as well as applications linked with events (Goeuriot, Mothe, Mulhem, Murtagh, & Sanjuan, 2016). One of the first pieces of information transmitted to disaster support systems is where the disaster has occurred (Lingad, Karimi, & Yin, 2013). A location within the text of a crisis message makes the message more valuable than messages that do not contain a location (Munro, 2011). In addition, Twitter users are most likely to pass on tweets with location and situational updates, indicating that Twitter users themselves find location to be very important (Vieweg, Hughes, Starbird, & Palen, 2010).

Name entity recognition in formal texts such as news and long documents has attracted many researchers. However, very little work has been successfully carried out on microblogs. The Stanford NER (Named Entity Recognition)[3] (Finkel, Grenager, & Manning, 2005) achieves an 89% F measure[4] for entity names on newswire, but only 49% for microblogs (Bontcheva et al., 2013). Similarly, the Gate NLP framework[5] (Bontcheva et al., 2013) achieves a 77% F measure for long texts but only 60% for short texts. The Ritter tool[6] (Ritter, Clark, & Etzioni, 2011), which is considered to be the state of the art, only achieves a 75% F measure for Twitter.

As mentioned in Bontcheva et al. (2013), each tool has its strengths and limitations. While the Gate NLP framework achieves high recall (83%) and low precision (47%), the Stanford NER achieves the opposite (recall 32%, precision 59%) for the development part of the Ritter dataset (Bontcheva et al., 2013).

Because there are applications that need high recall e.g. what has happened in a given location, and others that need high precision e.g. which locations should we concentrate on first for a given problem, we hypothesized that combining existing location extraction tools could improve the accuracy of location extraction. Moreover, we also hypothesized that filtering out the location using external resources could help the location extraction process. We thus derive our first research question:

RQ1: *How much can we improve precision and recall by combining existing tools to extract the location from microblog posts?*

To answer this question, we have combined various tools, namely, the Ritter tool (Ritter et al., 2011), the Gate NLP framework (Gate) (Bontcheva et al., 2013) and the Stanford NER (Finkel et al., 2005). We also propose to filter the extracted locations using DBpedia[7]. We have used it is as follows: the locations extracted by previous tools are only considered as locations if DBpedia considers them as locations (taking account of the DBpedia endpoint framework). We therefore targeted either recall oriented or precision oriented applications.

By associating locations that both Ritter and Gate recognize, we achieved 82% recall (for the Ritter dataset) which is very appropriate for recall oriented applications. This result can be explained by the fact that these methods use different clues to extract locations from tweets. On the other hand, when using DBPedia to filter out locations that Ritter recognizes, we reached a remarkable precision of 97% (for the Ritter dataset). This high result was obtained because imprecise recognized location names were discarded.

As mentioned earlier, social network and microblogs are widely used media of communication. As a result, a huge number of posts and tweets are posted daily, but only a very small proportion contains locations. For instance, in the Ritter dataset (Ritter et al., 2011), which was collected during September 2010, only about 9% of the tweets contain a location. It is thus time consuming to try to extract locations from texts where no location occurs. If we could filter out tweets that do not contain locations, *prior* to extracting locations, then efficiency would be improved. This leads us to our second research question:

RQ2: *Is it possible to predict whether a tweet contains a location or not?*

We conducted a preliminary study by using location extraction tools only on tweets that contain locations; we achieved sig nificantly higher accuracy than when implementing them on the entire datasets. This first result shows that if we could predict the fact that the text contains a location, it would be easier to extract this location.

One main contribution of this paper is that we define a number of new tweet features and use them as location predictors. Another contribution is that we evaluate the tweets using machine learning classifier algorithms with various parameters. In the experimental section, we show that the precision of NER tools for the tweets we predict to contain a location is significantly improved: from 85% to 96% for Ritter collection and from 80% to 89% for MSM2013 collection. This increase in precision is meaningful and crucial in systems where the location extraction needs to be very precise such as disaster supporting systems and rescues systems.

The rest of the paper is organized as follows: Section 2 presents the related work; Section 3 details the location extraction method we promote and its evaluation. In Section 4, we explain our original method to predict location occurrence in tweets and show its usefulness and effectiveness. Finally, Section 5 is the discussions and conclusion.

2. Related work

With the rising popularity of social media, many studies propose different ways to extract information from this resource. Previous similar studies can be grouped into two categories: location extraction and location prediction.

[3] http://nlp.stanford.edu/software/CRF-NER.shtml.

[4] F-measure is approximately the average (harmonic mean) of the precision and recall.

[5] https://gate.ac.uk/family/developer.html.

[6] https://github.com/aritter/twitter nlp.

[7] http://dbpedia.org/snorql/ BDpedia structures the information from Wikipedia pages; it can be queried using SPARQL to extract structured information locally stored in DBpedia or through an endpoint framework.

2.1. Location extraction

A piece of text related to a certain location includes information about that location. This information is either explicitly mentioned or inferred from the content. Identifying location names in a text is part of named entity recognition (NER). In information extraction it is a critical task for recognizing which parts of a text are mentioned as entity names.

In recent years, there have been a lot of NER systems that address the problem of extracting a location specified in documents (Bontcheva et al., 2013; Etzioni, 2005; Finkel et al., 2005; Kazama & Torisawa, 2008; Roberts, Gaizauskas, Hepple, & Guo, 2008); however they do not perform well on informal texts. The reason is probably because text parsers use features such as word type, capitalized letters and aggregated context, which are often not exact in noisy, unstructured, short microblogs (Huang, Liu, & Nguyen, 2015).

Previous studies on location identification rely mainly on: 1) searching and comparing the text for entity names in a gazetteer, and/or 2) using text structure and context. The former method is simple but limits the extraction to a predefined list of names, whereas the latter is able to recognize names even if they are not on the list (Huang et al., 2015).

Stanford NER is a very popular NER system. It applies a machine learning based method and is distributed with CRF models to detect named entities in English newswire text. Finkel et al. used simulated annealing in place of Viterbi coding in sequence models to enhance an existing CRF based system with long distance dependency models (Finkel et al., 2005). They outperform the NER in longer documents but do not perform well on microblogs as they achieve 89% for newswire but only 49% for tweets in the development of Ritter dataset (Bontcheva et al., 2013).

Agarwal et al. introduced an approach that combines the Stanford NER tool and a concept based vocabulary to extract location information from tweets. To filter out noisy terms from extracted location phrases, they used a naive Bayes classifier with the following features: the POS tags of the word itself, three words before this word, and three words after this word. To disambiguate place names, the authors extracted longitude and latitude information from a combination of an inverted index search on World Gazetteer data, and a search using Google Maps API (Agarwal, Vaithiyanathan, Sharma, & Shroff, 2012).

Kazama et al. introduced a method that uses large scale clustering of dependency relations between verbs and multi word nouns to build a gazetteer for detecting named entities in Japanese texts. They argue that, since the dependency relations capture the semantics on multi words, their cluster dictionary is a good gazetteer for NER. In addition, they also combined the cluster gazetteers with a gazetteer extracted from Wikipedia to improve accuracy (Kazama & Torisawa, 2008). Krishnan et al. presented a two stage method to deal with non local dependencies in NER (Krishnan & Manning, 2006) for long documents using Conditional Random Fields (CRF). Their first CRF based NER system used local features to make predictions while the second CRF was trained using both local information and features extracted from the output of the first CRF. This helped them build a rich set of features to model non local dependencies and conduct the inference efficiently since the inference time is merely one of two sequential CRFs. As a result, their method yielded a 12.6% relative error reduction on the F Measure, which is higher than the state of the art Stanford NER at 9.3%. Li and Sun (2014) extracted locations mentioned by Singapore users in their tweets. They built a location gazetteer by exploiting the crowdsourcing knowledge embedded in the tweets associated with Foursquare check ins. This inventory includes formal names and abbreviations commonly used to mention users' points of interest. When applying a linear chain CRF model that accounts for lexical, grammatical, and geographical features derived from the tweets and the gazetteer, the F measure for location recognition is about 8% higher than the Stanford NER. Ji, Sun, Cong, and Han (2016) reapplied the method from Li and Sun (2014) to address location recognition, which was a subtask in their work. This task is a sequential token tagging task applied according to the BILOU scheme in Ratinov and Roth (2009). As a result, they improved the F measure by about 0.05% compared to Li and Sun (2014).

Also applying CRF, but in a more complex way, Liu et al. combined a K Nearest Neighbors (KNN) classifier with a linear CRF model under a semi supervised learning framework to find named entities in tweets. They first used a KNN classifier to conduct word level classification, which exploits the similar, recently labeled tweets. These re labeled results, together with other conventional features, were then fed into the CRF model to capture fine grained information from a single tweet and from 30 gazetteers which cover common names, countries, locations and temporal expressions. By combining global evidence from KNN and the gazetteer with local contextual information, the researchers' approach was successful in dealing with the unavailability of training data (Liu, Zhang, Wei, & Zhou, 2011).

Li et al., in a different approach to previous studies, collectively identified named entities from a batch of tweets using an unsupervised method. Rather than relying on local linguistics features, they aggregated information garnered from the World Wide Web to construct local and global contexts for tweets. Firstly, they exploited on the global context retrieved from Wikipedia and the Web N Gram collection to segment microblogs. Each tweet segment was then considered as a candidate named entity. Next, they built a random model to exploit the gregarious property in the local context collected from the Twitter stream. The named entity is the highest ranked segment (Li et al., 2012). In another study, Ozdikis et al. determined the location of an event based on GPS geotags, tweet content and user profiles. They first separated these features and then combined them into a single solution using combination rules from DempsterShafer theory. On average, the city level error distance was 107,9 km (Ozdikis, Ouztzn, & Karagoz, 2016).

Recently, some approaches have been successful in detecting locations in tweets. Bontcheva et al. customized their NER systems for newswire, adapting the Gate NLP framework (Bontcheva et al., 2013) for tweets. They also adapted and retrained a Stanford tagger (Toutanova, Klein, Manning, & Singer, 2003) for tweet collections. They used gazetteers of personal names, cities and a list of unambiguous company and website names frequently mentioned in the training data. As a result, they increased the F measure from 60% to 80%, but mainly with respect to Person, Organization and Time, rather than Location.

Ritter et al. addressed the problem of NER for microblogs by using chunking to rebuild the NLP pipeline, beginning with part of speech tagging (Ritter et al., 2011). They applied a probabilistic model, LabelledLDA to exploit an open domain database (Freebase)

as a source of distant supervision. Their experiments show that their approach outperformed the existing NER tools on tweets for the Location entity type with a 77% F measure in finding location names in their own dataset, namely the Ritter dataset. While the Gate NLP framework achieves high recall, Stanford NER and Ritter are more efficient in terms of precision (Bontcheva et al., 2013). In this paper, we introduce a method that combines these tools to target either recall oriented or precision oriented applications. We also propose to filter the extracted locations using DBpedia to increase the precision of the tools.

2.2. Prediction of locations

Location prediction in tweets has been little studied. Recent work addressing this problem has followed two directions: content based and uncontent based. The first approach analyses the textual content while the second uses the information provided in user profiles, geo tagged tweets and social network information.

Wing et al. analyzed raw text to predict documents geo location in terms of latitude and longitude coordinates (Wing & Baldridge, 2011). They applied several supervised methods and used a geodesic grid as a discrete representation of the Earth's surface. Geo tagged documents were presented in a corresponding cell. New documents were geo located to the most similar cell based on Kullaback Leibler divergence (Zhai & Lafferty, 2001). Their prediction is impressive for Wikipedia articles with a median error of just 11.8 km; however, they do not perform well on tweets as the median error is 479 km.

Lee et al. developed a geo social event detection system by monitoring posts from Twitter users. They predicted the occurrence of events based on geographical regularities, which includes the three following indicators: the number of tweets, crowds and moving users, inferred from the usual behavior patterns of crowds with geo tag tweets. They compared these regularities with the estimated regularities to show the unusual events organized in the monitored geographical area. The sudden increase of tweets in a region and the increase of Twitter users in a short period of time are two important clues in their approach (Lee & Sumiya, 2010).

More recently, Ikawa et al. predicted the location where a message is generated by using its textual content. They derived associations between each location and its relevant keywords from past messages during the training and inferred where a new message comes from by comparing the similarities between the keywords in the training with the ones in the new message. They trained their datasets using two methods: for each user and for every user. They concluded that the training method for each user is more efficient in terms of recall and precision than the training method for every user (Ikawa, Enoki, & Tatsubori, 2012). Bo et al. predicted the geo location of a message or user based on the aggregated body of tweets from that user. They identified âÇ£location indicative wordsâÇ¥ (LIWs) that implicitly or explicitly encode an association with a particular location. They first detected IDWs via feature selection and then established whether the reduced feature set boosts geo location accuracy. Their results decreased the mean and median of the prediction error distance by 45 km and 209 km respectively (Bo, Cook, & Baldwin, 2012) .

In Backstrom, Sun, and Marlow (2010), the authors proposed an approach to predict the location of a user based on the user's friend. They modeled the relation between geographical distance and friendship and calculated the probability of a user being located at a specific place. The place with the maximum probability is estimated as the user location. As a result, they were able to estimate the location of 69% of users with 16 or more located friends to within 25 miles. Mahmud et al. infer the home location of Twitter users by extracting features from a user's tweets content and their tweeting behavior. They combined statistical and heuristic classifiers to predict locations and used a geography gazetteer to recognize location named entities (Mahmud, Nichols, & Drews, 2014). By using a user's profile and multiple map APIs, Kulshrestha et al. addressed the problem of finding a user's location at country level. They compared the location information obtained from multiple map APIs to reduce inference errors. Their approach was able to infer the location of 24% of users with 95% accuracy; however, it is not effective in cases where users input in correct information in the location field or leave it empty. Following this line of thought, Chandra, Khan, and Muhaya (2011) proposed a method of estimating the location of Twitter users, based purely on the content of the users' tweets along with the content of related reply tweets. They assumed that terms included in a user's tweets can be assigned as terms related to his or her town/city. Thus, they made use of a probabilistic framework that considers a distribution of terms found in the tweet messages from a specific dialogue, including reply tweets, initiated by the user. They also estimated the top K probable towns/cities for a given user and achieved the highest accuracy at 59% with K = 5, and an error distance of 300 miles.

Related studies focus on predicting the location of the users or where the text was generated, but not on predicting the occurrence of locations. Our study examines this prediction. The goal is to extract the smallest number of tweets that is likely to contain locations. If we are able to correctly predict the tweets in which a location is mentioned, we hypothesize that the precision and efficiency of NER tools can be improved since a very small proportion of tweets contain a location in their content.

In this paper, we rely on existing tools for location extraction and propose a method which predicts whether a tweet contains a location or not.

3. Combining location extraction methods

Name entity recognition (NER) in formal texts like news, documents has attracted many researchers. Location recognition is a part of the NER process in which locations are names of politically or geographically defined places such as regions, countries, cities, provinces, river and mountains. Locations also contain man made infrastructure such as airports, seaports, highways, streets and factories (for example: France, Asia, Vancouver and NY).

For Twitter, some approaches have been proposed and have been successful for location identification such as the Ritter tool (Ritter et al., 2011), the Gate NLP framework (Gate) (Bontcheva et al., 2013) and the Stanford NER (Finkel et al., 2005).

In this section, we focus on research question 1 ("How much can we improve precision and recall by combining existing tools?").

Table 1
Some features of the Ritter and MSM2013 datasets used to evaluate our location extraction and prediction models.

	Ritter's dataset	MSM2013 dataset
# of tweets	2394	2815
# of tweets containing a location (TCL)	213 (8.8%)	496 (17.6%)
# of tweets without location (TNL)	2181	2319

Table 2
Effectiveness when combining extraction models: Ritter, Gate, Stanford, and filtering with DBPedia. Recall - R(%), Precision - P(%), F-measure - F(%) for the Ritter and MSM2013 datasets. A statistically significant value is indicated by a star (*) when compared to the baseline.

	Ritter dataset			MSM2013 dataset		
	R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
Ritter (baseline)	71	82	77	61	80	69
Ritter + Stanford + DBp	77*	79	**78**	72*	79	**75***
Ritter + Gate + DBp	78*	71	74	74*	77	**75***
Ritter + Stanford	80*	64	72	**78***	72	**75***
Ritter + Gate	**82***	56	66	**78***	64	71
Ritter + DBp	45	**97***	62	48	**88***	62

We propose an approach to identify location names in tweets by combining these three tools and filtering out locations after extraction by DBPedia[8].

We first obtained the locations identified by each of the three tools. Then, we *merged* the extracted location names and finally we evaluated accuracy and precision.

To *filter* the locations, we checked their existence on a DBpedia endpoint framework which took account of the official name, abbreviation, postcode and nickname for the location and rejected location candidates not listed on DBpedia.

The results for recall, precision and F measure are shown in Table 2. We used the *t* test, with the entire dataset processed by the Ritter location extraction tool as the baseline (first row of Table 2).

We conducted experiments and evaluated our method for two public collections: the Ritter collection (Ritter et al., 2011) and the MSM2013 collection (Cano Basave, Varga, Rowe, Stankovic, & Dadzie, 2013), both of which are reference collections in the domain. The first collection was initially used by Ritter et al. (2011) while the second was the training dataset from Making Sense of Microposts 2013 (MSM2013). These two datasets are provided along with manual annotations on locations. Table 1 shows the number of tweets and their distribution (according to whether they mention a location or not) in both datasets.

As presented in Table 2, the combination of the Ritter location extraction tool and the Stanford NER filtered by DBpedia gives the best F measure, although only one percent higher than the baseline for the Ritter dataset. The F measure for the MSM2013 dataset has considerably increased with this combination (from 69% to 75%). The string of locations recognized by Ritter along with the locations identified by the Gate filtered by DBpedia (third row in Table 2) gives the second highest F measure for the Ritter dataset at 74% while the locations found by Ritter and Stanford (fourth row in Table 2) reach the F measure of 72%. These two combinations give the best results; an F measure of 75% for the SM2013 dataset.

Interestingly, we significantly improve recall in some cases and precision in others, which can be useful when both recall oriented or precision oriented applications are targeted.

Recall-oriented applications. The combination of Ritter and Gate gives the best recall, significantly increasing from 71% to 82% for the Ritter dataset while Ritter plus Stanford gives the second highest recall at 80% for the same dataset. The trend is similar for the MSM2013 dataset: the combination of Ritter with either Stanford or Gate gives the best recall at 78%, 17% higher than the baseline. As expected, precision is decreased in both combinations. Ritter combined with Stanford achieves a precision of 64% and 72% for the Ritter and MSM2013 datasets respectively while Ritter combined with Gate achieves 56% precision for Ritter dataset and 64% precision for the MSM2013 dataset. But overall, the F measure remains steady, even increasing in the case of MSM2013 dataset. These combinations can be applied to recall oriented applications such as Festival Recommender Systems, Entertainment Recommender Systems and Travel Recommender Systems since the recommendation is expected in as many places as possible.

Precision-orientated applications. Following our intuitive first idea to improve precision, we filtered out extracted locations by using DBpedia. As in the last row of Table 2, when locations identified by Ritter are filtered out, as expected, precision is greatly increased from 82% to 97% and from 80% to 88% for the Ritter and MSM2013 datasets respectively. However this improvement takes place to the detriment of recall: only 45% for the Ritter dataset and 48% for the MSM2013 dataset. This combination can be applied to precision oriented applications in which the precision is meaningful and essential, such as disaster support systems and rescue systems, although the F measure decreases.

[8] http://dbpedia.org/snorql/.

Table 3
Effectiveness of combining location extraction tools on Recall - R(%), Precision - P(%), F-measure - F(%) in tweets containing locations from the Ritter and MSM2013 datasets. A statistically significant value is indicated by a star (*) when compared to the baseline.

	Ritter dataset			MSM2013 dataset		
	R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
Ritter (baseline)	71	98	83	61	93	74
Ritter + Stanford + DBp	77*	95	**85**	72*	93	81*
Ritter + Gate + DBp	78*	95	85	74*	91	82*
Ritter + Stanford	80*	87	84	78*	89	**83***
Ritter + Gate	**82***	87	84	**78***	87	83*
Ritter + DBp	45	**99**	62	48	**96***	64

With regard to our first research question, we can conclude that combining Ritter and Gate is most appropriate in recall oriented applications since this combination significantly increases the recall of approximately 12% and 17% for the Ritter and MSM2013 datasets respectively. This may arise because these methods use different clues to extract locations in tweets. On the other hand, when precision is urgently required for precision oriented applications, the most effective method for filtering out locations recognized by Ritter is DBpedia: precision increases by 15% for the Ritter dataset and 8% for the MSM2013 dataset. As a good recall precision trade off, associating locations extracted by Ritter and Stanford filtered by DBpedia is sucessfil because it increases the F measure by 1% and 6% for the Ritter and MSM2013 datasets respectively.

4. Location prediction

In this section, we focus on the second research question: “Is it possible to predict whether a tweet contains a location or not?”. We also examine if this prediction is useful for location extraction accuracy. We first conducted a preliminary study to study the use fulness of location occurrence prediction by only applying prediction to tweets containing location and show that this is conclusive. We then proposed a model to predict the location occurrence in tweets and show the effectiveness of this model.

4.1. Location extraction on tweets containing locations

As a preliminary study of prediction usefulness, we conducted the same experiments as in Section 3 only for tweets containing locations. The objective was to see if it is more effective to extract locations from these tweets than from entire dataset. The results in terms of recall, precision and F measure are reported in Table 3. Overall, recall is unchanged but precision is greatly improved compared to the location extraction for the entire dataset. Interestingly, this leads to an increase in the F measure as well. As a baseline, Ritter tool leads to a sizeable increase in the F measure, from 77% to 83% and from 69% to 74% for the Ritter and MSM2013 datasets respectively.

The various combinations share the same general trend. When using DBPedia to filter out named entities extracted by Ritter (the last row of Table 3), we achieved the highest precision, 99% for the Ritter dataset and 96% for the MSM2013 dataset. The F measure is highest (85%) when combining Ritter with Stanford filtered by DBpedia for the Ritter dataset; the highest F measure for the MSM2013 dataset (83%) is also reached when combining Ritter with either Stanford or Gate.

From these results, it is obvious that using location extraction tools only on the tweets that contain locations, considerably improves precision, leading to an increase in the F measure. In addition, of the huge amount of tweets posted daily, there is a very small proportion of tweets that contains locations, therefore if we could exactly predict tweets that contain locations, we could filter out unnecessary tweets. This would save time and resources, and hopefully improve precision, which is essential and meaningful in precision oriented applications such as disaster support systems and rescue systems. This is why we have developed a model to predict whether a tweet contains a location or not; this model is presented in detail in the next sub section.

4.2. Predictive model

4.2.1. Tweet features

Predicting that a tweet contains a location name is not easy, as tweets are usually written in a pseudo natural language and may not correspond to grammatically correct sentences.

We manually analyzed some tweets from the festival tweet collection used in CLEF 2015 (Goeuriot et al., 2016) to detect clues that could be used to predict whether a location occurs in a tweet or not. We also relied on the related work regarding the pre positions that introduce a location.

Table 4 presents the features we propose along with some examples that support our choices. They are just examples, and some counter examples may exist, but we will revisit this aspect in the evaluation section.

The features “PP”,“Adj”,“Verb” are integers while the others are Yes/No values.

Geography gazetteer. This feature checks if a tweet contains at least one word appearing in a geography gazetteer. We chose the Gate NLP framework’s gazetteer which includes a list of countries, cities, regions and states with their abbreviations; it is available

Table 4
Features used to predict location occurrence in a tweet and examples of corresponding tweets.

Name	Description	Examples
1. Geography gazetteer	Contains a word appearing in Gate geography gazetteer	- Today I got a promotion at work , and tomorrow I 'm going home to **Wisconsin** for a few days.
2. Prep + PP	Contains a preposition just before proper nouns	- RT @RMBWilliams : Here **in Gainesville**! - Greek Festival **at St Johns** before ASPEN!
3. PP	Number of proper noun	going **to** alderwood :). # PP: 1
4.Prep	Contains one of the 7 prepositions London offices *on, from, to, toward, towards*	- Feeling really good after great week **in** our London offices - @Strigy got mine **in** bbt aintree today
5. Place + PP	Contains a word specifying place (*town, city, state, region, country*) just before or after proper noun	- The football fever : Ohio head coach Frank Solich says Ohio **state** knows they have a special team and season underway
6. Time	Contains a time expression (*today, tomorrow, weekend, tonight...*)	- Headed to da gump **today** alabama here I come - Come check out Costa Lounge **tonight**!
7. DefArt + PP	Contains a definite article just before proper noun	- Beautiful day! Nice to get away from **the Florida** heat
8. Htah	Contains a hashtag	**#Brazil**
9. Adj	Number of adjectives	- **Bad** time for leicester fans. # Adj:1
10. Verb	Number of verbs	- Willingham **took** a turn. # Verb: 2

online for open access and performs well in microblogs (Bontcheva et al., 2013).

As there is usually a preposition before a place name, we propose two features based on prepositions:

Prep. We define a binary feature to capture the presence of prepositions of place and movement[10](*at, in, on, from, to, toward, towards*).

Prep + PP. This feature checks if a tweet includes a preposition just before a proper noun (PP) and is recognized by Ritter POS.

Place + PP. This feature checks the presence of a specific word which often appear just after or just before a proper noun of place. We use the following words: *town, city, state, region, department and country*.

Time. We assume that a text about a specific place often includes a time expression. The time expressions checked included the words: *today, tomorrow, weekend, tonight*, the days of a week, and months.

DefArt + PP. The definite article "the" is used before country names such as *the Czech Republic, the United Arab Emirates* and *the United States* or before rivers, oceans, seas and mountain names. Thus, we define a binary feature that checks the presence of the following string type: "the" + PP.

Htah. Hashtag is one of the most ubiquitous aspects of Twitter. It is used to categorize tweets into topics. For events such as festivals or conferences, hashtag which specify the location of the events is widely used. This binary feature checks whether the tweet contains a hashtag or not.

PP, Adj, Verb. We count numbers of proper nouns, adjectives and verbs in a tweet recognized by the Ritter POS. We use these features in a predictive model that is derived using a training/testing framework.

We used the Ritter tool (Ritter et al., 2011), which is a state of the art POS in microblogs, to tag POS, and Python programing language to extract the features. The feature extraction processes took a few hours for each data collection on a computer with an i7 core processor and 16GB of RAM.

As reported later in sub Section 4.3.1, some features of the predictive model are more important than others and results may depend on optimized criteria (Section 4.3.2). Overall, we show that location extraction is more effective when applied to predicted tweets (Section 4.4).

Additionally, we evaluated our model using the Doc2vec model to infer vector features for representing tweets; however these features do not give good results for the prediction. The feature extraction as well as the results are detailed in Section 4.5.

4.2.2. Learning models and evaluation framework

We used the same collections as in Section 3 to evaluate our model: the Ritter dataset (Ritter et al., 2011) and MSM2013 dataset (Cano Basave et al., 2013) described in Table 1.

We tried different machine learning algorithms: the Naive Baiyes (NB), Support Vector Machine (SMO) and Random Forest (RF) using 10 folds cross validation. When training the model, it is possible to optimize various criteria. We consider that both accuracy and true positives should be optimized.

Machine learning algorithms also have some parameters. The so called "manual threshold" is a parameter for NB and RF clas sifiers and affects the prediction results. It corresponds to the statistically significant point which affects the output probability of the classifier. In our experiments, we varied the threshold (0.05, 0.20, 0.50, 0.75). On the other hand, SMO has an internal parameter called epsilon. This parameter is for the round off error. We varied epsilon (0.05, 0.20, 0.50, 0.75).

Baseline. We converted the content of tweets into word vectors classified by SMO (default setting) and considered it as baseline.

[9] http://grammar.ccc.commnet.edu/grammar/prepositions.htm.

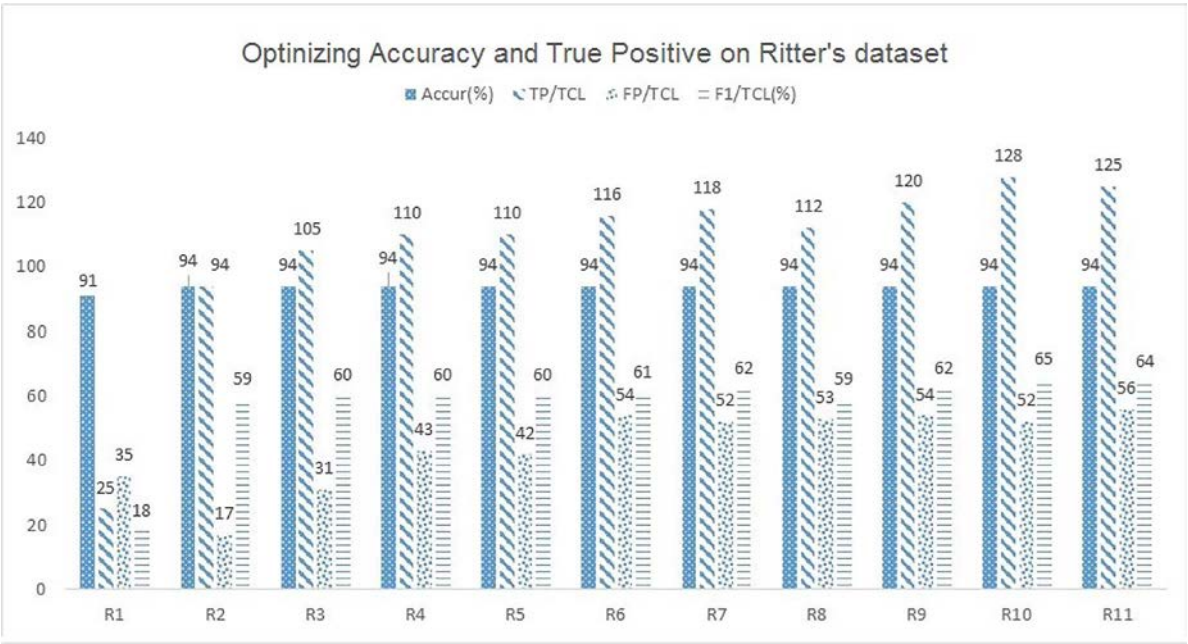

Fig. 1. Accuracy (%), true positive, false positive, and F-measure (%) for TCL (tweets containing a location) when optimizing *accuracy* and *true positive* obtained by a RandomForest threshold of 0.5 for the Ritter dataset with different numbers of features representing tweets.

All classification processes were implemented on a Weka graphical user interface (Hall et al., 2009). Some classifiers took longer than others, but all of them took a few minutes on a computer with an i7 core processor and 16 GB of RAM.

4.3. Results

4.3.1. The most important features for training

Our predictive model used 10 features, which were not all equally useful. We evaluated the importance of attributes by measuring the information gained with respect to the class. By setting the Inforgain attribute evaluator and the Ranker search method in Weka, we obtained the most important features, including the weight, as follows:

- Ritter's dataset: Geography gazetteer (0.145), Prep + PP (0.108), PP (0.0776), Pre + Place (0.02), Place + PP (0.002)
- MSM2013 dataset: Geography gazetteer (0.190), Prep + PP (0.093), Pre + Place (0.028), PP (0.023), DefArt + PP (0.005)

To evaluate how the results are improved after adding new features, we systematically combined features listed in Table 4 and ran additional experiments. For each run, we added one more feature. We started our experiments by running R1 including the first feature (Geography gazetteer) only. R2 consists of the first two features (Geography gazetteer and Prep + PP) while R3 contains the first three features (Geography gazetteer, Prep + PP and PP). The same rule was applied until all 10 features are included in the experiment which is R10. R11 was formed after removing features that decreased the results for runs from R1 to R10. R11 will be detailedly explained later in this section.

In Fig. 1 we present the results for accuracy (%), number of TP, FP and F measure (%) when optimizing accuracy and the true positive for the Ritter dataset (threshold 0.5) for all runs from R1 to R10 as described above. Logically, the best results are obtained at R10 which combines 10 features together.

When comparing the results for each run from R1 to R10 in Fig. 1, we can see that the F measure tends to increase as we add new features. There is one exception: the F measure for the R8 run decreases compared to the R7 run. Thus, we formed the R11 run including all features except the eighth feature "Hashtag" (see the ordered list in Table 4). However, the result for R11 is not higher than that for R10. We may suppose that the "Hashtag" might decrease the result for R8, but it may improve the result if combined with the ninth and tenth features, we therefore keep ten features.

Figs. 2 and 3 present the results for accuracy (%), number of TP, FP and F measure (%) for the R1 to R10 runs when optimizing accuracy and the true positive for the MSM2013 dataset respectively. Accuracy increases as we add new features to the model, while the F measure remains stable. The highest result when optimizing accuracy is obtained applying an RF threshold of 0.75 and when optimizing true positive applying an RF threshold of 0.2. From these two figures, we can see that some features have a reverse effect: these features increase the accuracy but decrease the true positive, for example, the R8 run is better than the R7 run when optimizing accuracy but lower when optimizing the true positive.

From the results above, we combined all 10 features for our later experiments.

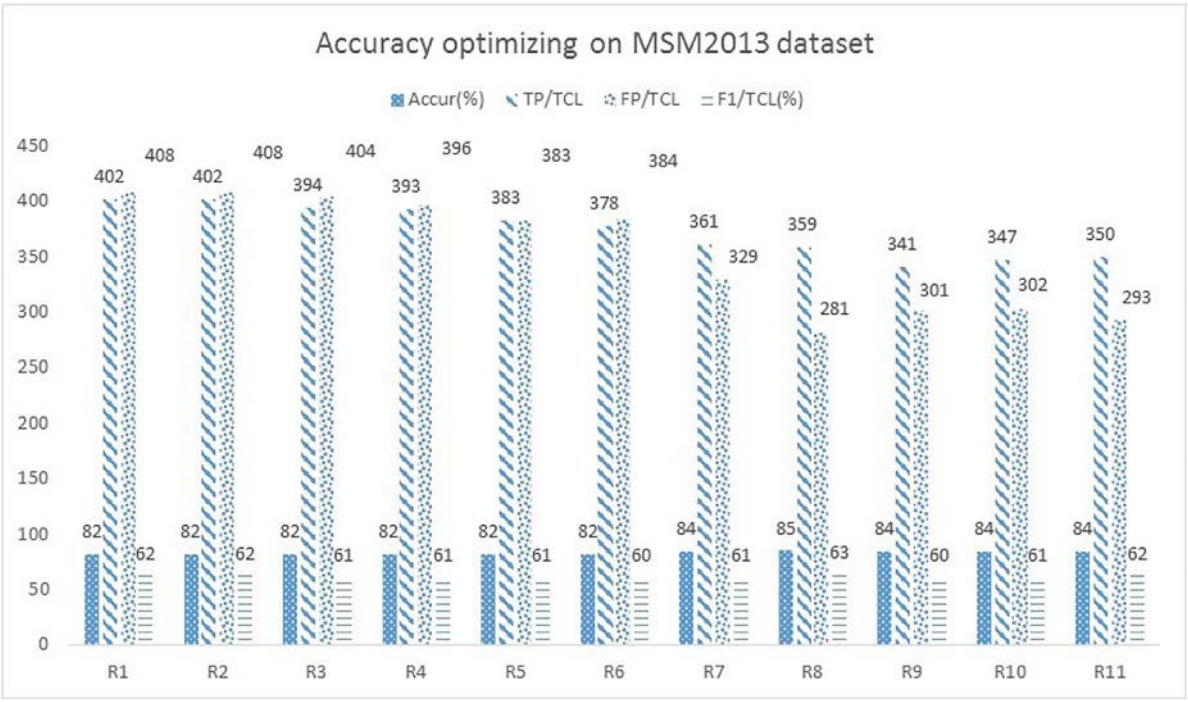

Fig. 2. Accuracy (%), true positive, false positive, and F-measure (%) for TCL when optimizing *accuracy* obtained by a RandomForest threshold of 0.75 for the MSM2013 dataset with different numbers of features representing tweets.

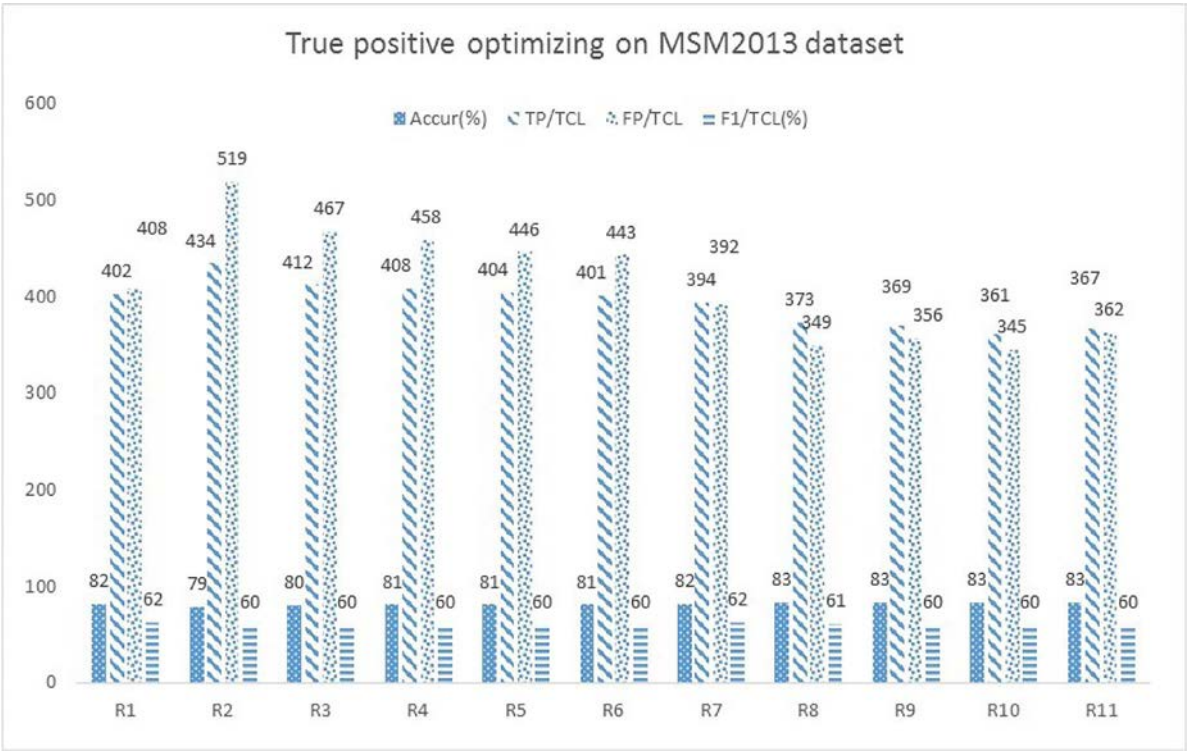

Fig. 3. Accuracy (%), true positive, false positive, and F-measure (%) for TCL when optimizing *true positive* obtained by a Randomforest threshold of 0.2 for the MSM2013 dataset with different numbers of features representing tweets.

Table 5
Accuracy (Acc - %), true positive (TP), false positive (FP), and F-measure (F-%) for TCL when optimizing either *accuracy* or *true positives* - 10-folds cross validation when using Naive Bayes (NB) and Random Forest (RF) for both collections. The number next to the ML algorithm indicates the threshold used. The number next to TP is the percentage of TP obtained out of the TCL while the number next to FP is the percentage of FP obtained out of TNL.

		Ritter dataset				MSM2013 dataset			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Baseline	SMO (1e−12)	92	36(17)	8(0.4)	28	87	184(37)	50(2.2)	50
Acc	SMO (1e−12)	94	99 (47)	21 (1.0)	60	88	226 (46)	61 (3.0)	58
Acc	NB (0.75)	90	153 (72)	177 (8.0)	56	82	357 (72)	375 (16)	58
Acc	RF (0.75)	92	152 (71)	133 (6.0)	61	**84**	**347 (70)**	**302 (13)**	**61**
Acc	NB (0.5)	92	129 (61)	96 (4.0)	59	89	236 (48)	107 (5.0)	56
Acc	RF (0.5)	**94**	**128 (60)**	**52 (2.0)**	**65**	87	263 (53)	130 (6.0)	59
TP	SMO (1e−12)	94	99 (47)	21 (1.0)	59	88	22 (4.0)	61 (3.0)	58
TP	SMO (0.05)	93	133 (62)	97 (4.0)	60	86	267 (54)	160 (7.0)	50
TP	SMO(0.2)	92	137 (64)	124 (6.0)	58	82	327 (66)	350 (15)	56
TP	SMO(0.5)	86	132 (62)	253 (12)	44	76	325 (66)	509 (22)	49
TP	SMO(0.75)	91	0 (0.0)	0 (0.0)	0.0	82	0.0 (0.0)	0.0 (0.0)	0.0
TP	NB (0.05)	*86*	*190 (89)*	*319 (15)*	*53*	*74*	*450 (91)*	*685 (30)*	*55*
TP	NB (0.2)	89	160 (75)	203 (9.0)	56	80	400 (81)	472 (20)	59
TP	NB (0.5)	92	129 (61)	96 (4.0)	59	87	236 (48)	107 (5.0)	56
TP	NB (0.75)	93	119 (56)	69 (3.0)	59	87	183 (37)	40 (2.0)	51
TP	RF(0.05)	84	181 (85)	341 (16)	49	70	428 (86)	781 (34)	50
TP	RF(0.20)	91	158 (74)	164 (8.0)	59	**83**	**361 (73)**	**345 (15)**	**60**
TP	RF(0.5)	**94**	**128 (60)**	**52 (2.0)**	**65**	87	263 (53)	130 (6.0)	59
TP	RF(0.75)	94	84 (39)	20 (1.0)	53	87	188 (38)	49 (2.0)	51

4.3.2. Optimized criteria

Table 5 presents the results for the various machine learning models. The rows in the first part of the table report the results when accuracy is optimized, while the second part reports the results when the number of true positives (TP) is optimized. The second column reports the results for the Ritter dataset while the third column reports the results for the MSM2013 dataset. The rows in bold highlight the best F measure while the rows in italic highlight the highest true positive score obtained.

The best F measure (65%) for the Ritter dataset is obtained using a RF with threshold of 0.5 (second row, Ritter column in Table 5). Prediction accuracy is 94% with 128 TP for 213 tweets containing a location TCL (60%), 52 false positives (FP) over 2.181 tweets not containing a location (TNL) (2%) when optimizing accuracy. When optimizing TP, the same configuration achieves the best results in terms of the F measure.

This configuration is second best only when applied to the MSM2013 dataset (F measure 59%). For this collection, the highest F measure when optimizing accuracy is obtained for a RF threshold of 0.75 (61% F measure). When optimizing TP the best threshold for RF is 0.2 (F measure 60%). Interestingly, NB with a threshold of 0.05 achieves an impressive TP for both collections although the number of FP increases. We obtain 190TP/213TCL (89%) and 319FP/2181TNL (15%) for the Ritter collection compared to 450TP/496TCL (91%) and 685FP/2319TNL (30%) for the MSM2013 collection.

SMO gives the highest accuracy, but not the best F measure (for TCL) or TP relative to RandomForest and Naive Bayes, which are presented in Table 5.

For the Ritter dataset, accuracy is from 84% to 94%; it is a little lower for the MSM2013 dataset but still higher than 80% in most cases. When calculating accuracy, both the predicted TCL and TNL are considered, although we are more interested in the correct prediction for TCL. This is why Table 5 also reports the results for TCL: true positive, false positive and the F measure.

Optimizing the TP criteria rather than accuracy leads to different TP results although the F measure does not change much apart from the RF model.

To sum up our findings, applying RF with a threshold of 0.5 gives the best F measure at 65% for the Ritter dataset when optimizing both accuracy and TP, this configuration achieves the second best F measure for the MSM2013 dataset, which is 2% lower than the best F measure when optimizing accuracy (using a RF threshold of 0.75) and 1% lower than the best F measure when optimizing TP (using a RF threshold of 0.2).

4.4. Location extraction for predicted tweets

We showed in Sections 4.2 and 4.3 that it is possible to train a model to predict if a tweet contains a location. Table 7 presents the results we obtained when extracting locations from those predicted tweets. We report the results both on predicted TCL and the results when the test sets are used (the details of test sets are explained below). We used three draws and report the average numbers. The number in brackets is the best result from the three draws.

Statistical significance is marked by a *. We used the *t* test considering the entire testing data set treated by the Ritter location extraction tool as the baseline (first row Table 7). When several draws were used, the individual significance of each draw was calculated and a * means that the difference with the baseline is statistically significant for the three draws. The training and testing

Table 6
Description of data used for training and testing.

	Ritter's dataset	MSM2013 dataset
Training	142 TCL, 1420 TNL	331 TCL, 1655 TLN
Testing	71 TCL, 761 TNL	165 TCL, 664 TNL

Table 7
Effectiveness of the Ritter algorithm for the Ritter and MSM2013 data collections in terms of Recall, Precision, F-measure, considering the entire testing set as described in Table 6 and the tweets we predict as containing a location. A statistically significant value is indicated by a star (*) when compared to the baseline. The number in brackets is the best result from the three draws.

		Ritter dataset			MSM2013 dataset		
		R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
Baseline	Entire testing set	69	85	75	60	80	69
Accuracy	TCL predicted by RF (0.5)	45(51)	**96*(98)**	61(66)	37(40)	**89*(92)**	52(55)
Accuracy	TCL predicted by RF (0.75)	53(58)	**92*(96)**	67(68)	46(48)	**86*(88)**	60(61)
TP	TCL predicted by RF (0.2)	56(63)	**91*(96)**	69(71)	49(51)	**87*(88)**	63(64)
TP	TCL predicted by RF (0.5)	45(51)	**96*(98)**	61(66)	37(40)	**89*(92)**	52(55)
TP	TCL predicted by NB (0.05)	64(69)	**88(93)**	74(75)	58(61)	**82(85)**	68(70)

sets were built from the Ritter and MSM2013 collections following the principle that the unbalanced nature of the data set was set; 2/3 of TCL are used for training and 1/3 for testing. Exact numbers are provided in Table 6.

As in Table 7, precision significantly increases for both Ritter and MSM2013 collections from 85% to 96% and from 80% to 89% respectively; although recall decreases due to the errors caused by filtering tweets with BDpedia; specifically because abbreviations of locations are usually not mentioned in this resource.

A high precision is important in precision oriented applications. In addition, by running NER tools only on the tweets that are predicted to contain a location, we can save time and resource compared to running these tools on the complete original collections.

4.5. Applying Doc2Vec to location prediction

In addition to the features of our model mentioned in Table 4, we tried to build other vector features using the Doc2Vec model (Le & Mikolov, 2014). We hypothesized that tweets about a given location will somehow relate to each other. For instance, consider the following two tweets: *"Vietnam, what a cool country to visit!!!"* and *"Valras, that was cool"* . Intuitively, these two tweets do not seem to "relate" to each other, but since they share some words in sentence structure and Vietnam is obviously a location, we can infer that Valras is also a location.

Following that idea, we tried to represent tweets as vectors and used these vectors as features to classify tweets according to whether they contain a location or not. Tweets which have similar vectors should be in the same class.

We used the document vector (Doc2Vec) model, which is "an unsupervised framework that learns continuous distributed vector representations for pieces of texts" Le and Mikolov (2014) trained on different large datasets to infer vector for tweets in the two collections we used previously: Ritter and MSM2013. These vectors are used in turn as features for the classification model as presented in Section 4.2, with the same classifier algorithms and parameters. We chose this model because Doc2Vec is considered as an efficient tool to compute vectors representing documents and has recently been applied in various research areas. We believe that if we were to use a sufficiently large and appropriate training dataset which covers information on locations around the world, we could infer appropriate vector representations that could lead to better location prediction.

We respectively trained the Doc2vec model on three different datasets as follows:

- English Wikipedia (Lau & Baldwin, 2016) which is dump dated 2015 12 01 including approximately 35 million documents.
- English tweets (Iso language code "en") of CLEF festival dataset (Goeuriot et al., 2016) which is collected from June to September 2015, including 9,073,707 tweets.
- English tweets of 1 percent tweets collection which was collected from September 2015 to October 2016, composed of 21,634,176 tweets.

When trained on the above three datasets, the Doc2Vec model is configured using the following hyper parameter values: the dimensionality of feature vectors size = 300, the initial learning rate alpha = 0.025, the number of core machine used for this process workers = 6, takes into consideration the words with total frequency at least min count = 3. The other parameters are set as default.

We respectively ran location prediction experiments using the features described below. The other settings (algorithms and parameters) are the same as in Section 4.2.

Table 8
Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F%) for TCL when optimizing either *accuracy* or *true positives* - 10-folds cross validation when using features: vectors inferred from the Doc2Vec model trained on the English Wikipedia collection, mean, max, min and standard deviation of these inferred vectors.

		Ritter dataset				MSM2013 dataset			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e−12)	91	68	61	40	86	218	106	53
Acc	NB (0.75)	77	133	479	32	77	317	473	49
Acc	RF (0.75)	92	44	32	30	82	305	328	54
Acc	NB (0.5)	78	128	449	32	78	292	425	48
Acc	RF (0.5)	91	0	0	0	83	29	3	11
TP	SMO (1e−12)	94	68	61	40	86	218	106	53
TP	SMO (0.05)	86	97	211	37	77	312	462	49
TP	SMO(0.2)	79	119	408	32	76	311	479	48
TP	SMO(0.5)	83	43	240	17	70	132	467	24
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	74	143	548	32	74	342	583	48
TP	NB (0.2)	76	136	490	32	76	324	497	49
TP	NB (0.5)	78	128	449	32	77	292	425	48
TP	NB (0.75)	79	121	415	32	79	271	373	48
TP	RF(0.05)	43	200	1348	23	25	487	2096	31
TP	RF(0.20)	89	71	114	36	75	375	591	51
TP	RF(0.5)	91	0	0	0	83	29	3	11
TP	RF(0.75)	91	0	0	0	82	0	0	0

- **Run 1**. The features are vectors inferred from the Doc2Vec model trained on the English Wikipedia collection, mean, max, min and standard deviation of these vectors. The results for location prediction are reported in Table 8.
- **Run 2**. The features are vectors inferred from the Doc2Vec model trained on the English Wikipedia collection, mean, max, min and standard deviation of these vectors, plus 10 features mentioned in Table 4. The results of location prediction are reported in Table 9.
- **Run 3**. The features are vectors inferred from the Doc2Vec model trained on the CLEF festival collection, mean, max, min and standard deviation of these vectors. The results of location prediction are reported in Table 10.
- **Run 4**. The features are vectors inferred from the Doc2Vec model trained on the CLEF festival collection, mean, max, min and standard deviation of these vectors, plus 10 features mentioned in Table 4. The results of location prediction are reported in Table 11.
- **Run 5**. The features are vectors inferred from Doc2Vec model trained on the 1 percent tweet collection, mean, max, min and standard deviation of these vectors. The results of location prediction are reported in Table 12.

Table 9
Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F -%) for TCL when optimizing either *accuracy* or *true positives* - 10-folds cross validation when using features: vectors inferred from the Doc2Vec model trained on the English Wikipedia collection, mean, max, min and standard deviation of these inferred vectors plus 10 features mentioned in Table 4.

		Ritter dataset				MSM2013 dataset			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e−12)	**93**	**126**	**70**	**62**	**89**	**314**	**126**	**67**
Acc	NB (0.75)	79	144	429	37	79	336	418	54
Acc	RF (0.75)	93	105	58	56	87	367	239	67
Acc	NB (0.5)	80	140	401	37	80	319	374	54
Acc	RF (0.5)	91	0	0	0	85	90	3	31
TP	SMO (1e−12)	**93**	**126**	**70**	**62**	**89**	**314**	**126**	**67**
TP	SMO (0.05)	92	149	127	61	84	365	323	62
TP	SMO(0.2)	91	156	165	58	84	359	309	62
TP	SMO(0.5)	89	29	87	18	79	54	137	16
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	77	158	487	37	77	357	510	52
TP	NB (0.2)	78	146	439	37	79	338	432	53
TP	NB (0.5)	80	140	401	37	80	319	374	54
TP	NB (0.75)	81	133	372	37	81	298	348	60
TP	RF(0.05)	53	211	1127	27	29	493	1987	33
TP	RF(0.20)	92	138	119	59	79	406	493	58
TP	RF(0.5)	91	0	0	0	85	90	3	31
TP	RF(0.75)	91	0	0	0	82	0	0	0

Table 10
Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F-%) for TCL when optimizing either *accuracy* or *true positives* - 10-folds cross validation when using features: vectors inferred from the Doc2Vec model trained on the CLEF festival collection, mean, max, min and standard deviation of these inferred vectors.

		Ritter dataset				MSM2013 dataset			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e−12)	91	10	22	8.0	84	106	65	32
Acc	NB (0.75)	73	86	523	21	71	234	552	37
Acc	RF (0.75)	91	4	8	4.0	78	214	340	41
Acc	NB (0.5)	76	75	434	21	74	197	444	35
Acc	RF (0.5)	91	0	0	0	82	2	2	1
TP	SMO (1e−12)	91	10	22	8.0	84	106	65	32
TP	SMO (0.05)	81	86	338	27	79	204	291	41
TP	SMO(0.2)	77	84	418	24	67	286	719	38
TP	SMO(0.5)	87	18	123	10	67	185	614	29
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	67	102	687	20	66	283	748	37
TP	NB (0.2)	72	86	543	20	70	244	588	37
TP	NB (0.5)	76	75	434	21	74	197	444	35
TP	NB (0.75)	79	67	361	21	75	163	366	32
TP	RF(0.05)	27	185	1709	18	20	492	2248	30
TP	RF(0.20)	90	17	54	12	69	316	695	42
TP	RF(0.5)	91	0	0	0	82	2	2	1
TP	RF(0.75)	91	0	0	0	82	0	0	0

Table 11
Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F-%) for TCL when optimizing either *accuracy* or *true positives* - 10-folds cross validation when using features: vectors inferred from the Doc2Vec model trained on the CLEF festival collection, mean, max, min and standard deviation of these inferred vectors plus 10 features mentioned in Table 4.

		Ritter dataset				MSM2013 dataset			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e−12)	93	124	72	61	**89**	**307**	**109**	**67**
Acc	NB (0.75)	80	121	398	57	77	280	449	46
Acc	RF (0.75)	94	84	31	51	86	346	254	63
Acc	NB (0.5)	82	111	333	33	78	249	362	45
Acc	RF (0.5)	91	0	0	0	83	15	1	6
TP	SMO (1e−12)	93	124	72	61	**89**	**307**	**109**	**67**
TP	SMO (0.05)	89	151	208	53	85	323	243	61
TP	SMO(0.2)	90	135	169	52	84	358	300	62
TP	SMO(0.5)	91	15	19	12	74	91	336	20
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	75	137	524	31	72	321	620	45
TP	NB (0.2)	79	124	418	33	76	285	472	46
TP	NB (0.5)	82	111	333	34	79	249	362	45
TP	NB (0.75)	84	94	275	32	80	213	277	43
TP	RF(0.05)	45	206	1305	24	24	493	2139	32
TP	RF(0.20)	92	122	105	56	77	399	546	55
TP	RF(0.5)	91	0	0	0	83	15	1	6
TP	RF(0.75)	91	0	0	0	82	0	0	0

- **Run 6**. The features are vectors inferred from the Doc2Vec model trained on the 1% tweet collection, mean, max, min and standard deviation of these vectors, plus 10 features mentioned in Table 4. The results of location prediction are reported in Table 13.

Our intuition when applying a model to represent tweets as vectors and predict location occurrence in tweets based on the similarity of vectors has not been confirmed by the results. We achieved lower F measure in almost all cases, except for the 62% and 67% when applying SMO (epsilon 1e 12, both accuracy and true positive optimizing) for the Ritter and MSM2013 data collection respectively (the first and sixth rows in Table 9) using vectors inferred from the Doc2Vec model trained on the English Wikipedia collection combining with 10 features mentioned in Table 4. We also achieved the highest F measure when applying this config uration to the MSM2013 dataset using vectors inferred from the Doc2Vec model trained on the CLEF festival collection combined

Table 12
Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F-%) for TCL when optimizing either *accuracy* or *true positives* - 10-folds cross validation when using features: vectors inferred from the Doc2Vec model trained on the 1Ptweets dataset, mean, max, min and standard deviation of these inferred vectors.

		Ritter dataset				MSM2013 dataset			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e−12)	91	20	33	15	83	107	79	31
Acc	NB (0.75)	68	92	649	19	67	223	659	32
Acc	RF (0.75)	91	6	9	5.3	76	197	390	36
Acc	NB (0.5)	69	85	605	19	69	196	567	31
Acc	RF (0.5)	91	0	0	0	82	0	1	0
TP	SMO (1e−12)	91	20	23	15	69	196	567	31
TP	SMO (0.05)	78	97	401	27	74	238	473	39
TP	SMO(0.2)	73	95	518	23	64	283	781	36
TP	SMO(0.5)	85	29	171	14	66	161	613	25
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	64	110	761	20	61	275	866	34
TP	NB (0.2)	67	95	663	20	66	229	629	32
TP	NB (0.5)	69	85	605	19	69	196	567	31
TP	NB (0.75)	71	77	547	18	71	163	477	29
TP	RF(0.05)	33	197	1590	20	22	490	2201	30
TP	RF(0.20)	89	26	77	17	65	303	770	39
TP	RF(0.5)	91	0	0	0	82	0	1	0
TP	RF(0.75)	91	0	0	0	82	0	0	0

Table 13
Accuracy (Acc - %), true positive (TP), false positive (FP), and the F-measure (F-%) for TCL when optimizing either *accuracy* or *true positives* - 10-folds cross validation when using features: vectors inferred from the Doc2Vec model trained on the 1Ptweets, mean, max, min and standard deviation of these inferred vectors plus 10 features mentioned in Table 4.

		Ritter dataset				MSM2013 dataset			
Optimize	ML (parameter)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)	Acc (%)	TP ($\frac{TP}{TCL}$%)	FP ($\frac{FP}{TNL}$%)	F (%)
Acc	SMO (1e−12)	93	118	78	58	87	261	125	59
Acc	NB (0.75)	74	123	532	28	72	260	560	40
Acc	RF (0.75)	93	85	33	51	84	333	276	60
Acc	NB (0.5)	75	110	488	27	74	237	470	39
Acc	RF (0.5)	91	1	0	0	83	13	1	5
TP	SMO (1e−12)	93	118	78	58	87	261	125	59
TP	SMO (0.05)	89	144	185	53	84	335	287	60
TP	SMO(0.2)	89	149	190	54	83	307	288	56
TP	SMO(0.5)	91	15	19	12	76	62	229	16
TP	SMO(0.75)	91	0	0	0	82	0	0	0
TP	NB (0.05)	70	132	630	27	67	314	701	42
TP	NB (0.2)	73	124	552	28	71	269	581	40
TP	NB (0.5)	75	110	488	27	74	237	470	39
TP	NB (0.75)	77	104	444	27	75	208	413	37
TP	RF(0.05)	44	207	1322	24	25	492	2101	32
TP	RF(0.20)	92	130	103	58	77	414	568	56
TP	RF(0.5)	92	1	0	0	83	13	1	5
TP	RF(0.75)	91	0	0	0	82	0	0	0

with the 10 features mentioned in Table 4. We suppose that the main reason for the prediction failure is the quality of the datasets used for training the Doc2Vec model. Although, the English Wikipedia collection covers information related to locations around the world, it includes documents and structured texts written in formal language. Thus, when applied to noisy, short, unstructured texts such as tweets, the inferred vectors are not exact. Besides, the 1 percent tweets collection is randomly collected from Twitter which might contain very little information related to locations while the CLEF festival collection is more about events than locations and may not be large enough.

Although, we have not been successful when using inferred vectors from the Doc2Vec model trained on different data collections, we believe that we could achieve better results if we had a “good” enough training dataset for the Doc2Vec model covering in formation related to locations around the world; but this question will have to be left for a future work.

5. Conclusion and discussion

In this paper, we have proposed an approach for location extraction and a model to predict the location occurrence in tweets, these results have direct applications on retweetability prediction (Hoang & Mothe, 2017). Our approach for location extraction is based on the combination of existing location extraction methods and significantly improves performance when we target either recall or precision oriented applications. We have shown that:

(1) Combining locations recognized by the Ritter tool with locations recognized by Stanford filtered by DBpedia increases the F measure for location extraction.
(2) Combining the locations extracted by Ritter with locations recognized by Gate considerably improves recall while using DBpedia to filter out location entities recognized by Ritter remarkably increases precision.
A vast amount of tweets are posted daily however very little proportion of them contains locations. In addition, running location extraction tools only on the tweets that contain locations significantly improves the results. We hypothesized that we could greatly increase the precision if we could predict the location occurrence in tweets. We thus introduced a method to predict whether a tweet contains a location or not. We defined some new features to represent tweets and intensively evaluate machine learning settings to predict location occurrence by varying the machine learning algorithm and the machine learning parameters used. The results show that:
(3) Random Forest and Naive Baiyes are the best machine learning solutions for this problem they perform better than Support Vector Machine (and other algorithms we tried but did not report).
(4) Changing the criteria to optimize (accuracy or true positive) does not change the F measure much while it has an impact on true positive and false positive.
(5) When considering location extraction, we improved precision by focusing only on the tweets that are predicted as containing a location.

Our model gives an exact prediction for tweets that contain words from the geography gazetteer or include a preposition just before a proper noun. We also obtained a good prediction on tweets based on 'number of proper nouns' or 'words specifying places just after or before proper noun'. However, we have some cases where prediction is not appropriate. Since we only considered the abbreviations of locations included in the Gate frameworks gazetteer, some tweets are not predicted accuratly if they mention abbreviations not included in the gazetteer such as: *"@2kjdream Good morning ! We are here JPN!"* where JPN is not recognized ?. Besides, we have not dealt with location disambiguation. We believe that for future work and in order to solve this problem, the context given by all the words in the message should be considered (SanJuan, Moriceau, Tannier, Bellot, & Mothe, 2012).

In addition, our attempts to improve the results using word embedding representations for tweets were not successful; we believe this might be due to the non appropriate training collections available to date.

In future work, we will build relevant training datasets for the Doc2Vec to infer vector features representing tweets. We think that appropriate training datasets will overcome the limitations of our model i.e. abbreviations and disambiguation. Tweets that contain similar words about the same stories or events should be about the same locations. We also plan to extract more features to improve our predictive model. Finally, while this paper has focused on locations, we would also like to define predictive models for other types of entities such as people names.

References

Agarwal, P., Vaithiyanathan, R., Sharma, S., & Shroff, G. (2012). *Catching the long-tail: Extracting local news events from twitter. ICWSM*.

Backstrom, L., Sun, E., & Marlow, C. (2010). *Find me if you can: Improving geographical prediction with social and spatial proximity. Proceedings of the 19th international conference on world wide web*. ACM61–70.

Bo, H., Cook, P., & Baldwin, T. (2012). *Geolocation prediction in social media data by finding location indicative words. Proceedings of COLING*1045–1062.

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., & Aswani, N. (2013). *TwitIE: An open-source information extraction pipeline for microblog text. RANLP*83–90.

Cano Basave, A. E., Varga, A., Rowe, M., Stankovic, M., & Dadzie, A. S. (2013). *Making sense of microposts (# msm2013) concept extraction challenge. Proceedings of the concept extraction challenge at the workshop on 'Making Sense of Microposts' co-located with the 22nd international World Wide Web conference (WWW'13)*.

Chandra, S., Khan, L., & Muhaya, F. B. (2011). *Estimating twitter user location using social interactions-a content based approach. Privacy, security, risk and trust (PASSAT) and 2011 IEEE third inernational conference on social computing (socialcom), 2011 IEEE Third International Conference on*838–843.

Etzioni, O., et al. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence, 165*(1), 91–134.

Finkel, J. R., Grenager, T., & Manning, C. (2005). *Incorporating non-local information into information extraction systems by gibbs sampling. Proceedings of the 43rd annual meeting on association for computational linguistics. association for computational linguistics*363–370.

Goeuriot, L., Mothe, J., Mulhem, P., Murtagh, F., & Sanjuan, E. (2016). *Overview of the CLEF 2016 cultural micro-blog contextualization workshop. In experimental IR meets multilinguality, multimodality, and interaction. Proceedings of the seventh international conference of the CLEF association (CLEF 2016), Lecture Notes in Computer Science (LNCS) 9822*. Springer, Heidelberg, Germany371–378. http://dx.doi.org/10.1007/978-3-319-44564-9-30.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter,* 10–18.

Hoang, T. B. N., & Mothe, J. (2017). Predicting information diffusion on Twitter-Analysis of predictive features. *Journal of Computational Science*. http://dx.doi.org/10.1016/j.jocs.2017.10.010.

Huang, Y., Liu, Z., & Nguyen, P. (2015). *Location-based event search in social texts. Computing, networking and communications (ICNC), 2015 international conference on*668–672.

Ikawa, Y., Enoki, M., & Tatsubori, M. (2012). *Location inference using microblog messages. Proceedings of the 21st international conference on world wide web*. ACM687–690.

Ji, Z., Sun, A., Cong, G., & Han, J. (2016). *Joint recognition and linking of fine-grained locations from tweets. Proceedings of the 25th international conference on world wide web. International world wide web conferences steering committee*1271–1281.

Kazama, J. I., & Torisawa, K. (2008). *Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. Proceedings of ACL-08: HLT*407–415.
Krishnan, V., & Manning, C. D. (2006). *An effective two-stage model for exploiting non-local dependencies in named entity recognition. The 21st international conference on computational linguistics and the 44th annual meeting of the association for computational linguistics*1121–1128.
Lau, J. H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *Proceedings of the 1st Workshop on Representation Learning for NLP* (pp. 78–86). Berlin, Germany Arxiv preprint1607.05368
Le, Q., & Mikolov, T. (2014). *Distributed representations of sentences and documents. Proceedings of the 31st international conference on machine learning (ICML-14)* 1188–1196.
Lee, R., & Sumiya, K. (2010). *Measuring geographical regularities of crowd behaviors for twitter–based geo-social event detection. Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks*1–10.
Li, C., & Sun, A. (2014). *Fine-grained location extraction from tweets with temporal awareness. Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval*. ACM43–52.
Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., & Lee, B. S. (2012). *Twiner: Named entity recognition in targeted twitter stream. The 35th international ACM SIGIR conference on research and development in information retrieval*721–730.
Lingad, J., Karimi, S., & Yin, J.. (2013). *Location extraction from disaster-related microblogs. Proceedings of the 22nd international conference on world wide web*. ACM1017–1020.
Liu, X., Zhang, S., Wei, F., & Zhou, M. (2011). *Recognizing named entities in tweets. Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*359–367.
Mahmud, J., Nichols, J., & Drews, C. (2014). *Home location identification of twitter users. ACM transactions on intelligent systems and technology (TIST)*.
Munro, R. (2011). *Subword and spatiotemporal models for identifying actionable information in haitian kreyol. Proceedings of the fifteenth conference on computational natural language learningAssociation for Computational Linguistics*68–77.
Ozdikis, O., Ouztzn, H., & Karagoz, P. (2016). Evidential estimation of event locations in microblogs using the dempstershafer theory. *Information Processing & Management, 52*(6), 1227–1246.
Ratinov, L., & Roth, D. (2009). *Design challenges and misconceptions in named entity recognition. Proceedings of the thirteenth conference on computational natural language learningAssociation for Computational Linguistics*147–155.
Ritter, A., Clark, S., & Etzioni, O. (2011). *Named entity recognition in tweets: An experimental study. Proceedings of the conference on empirical methods in natural language processing. association for computational linguistics*.
Roberts, A., Gaizauskas, R. J., Hepple, M., & Guo, Y. (2008). *Combining terminology resources and statistical methods for entity recognition: An evaluation. LREC*.
SanJuan, E., Moriceau, V., Tannier, X., Bellot, P., & Mothe, J. (2012). *Overview of the INEX 2012 tweet contextualization track. Initiative for XML retrieval INEXvol. 148*.
Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). *Feature-rich part-of-speech tagging with a cyclic dependency network. Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology-volume 1Association for Computational Linguistics*173–180.
Vieweg, S., Hughes, A. L., Starbird, K., & Palen, L. (2010). *Microblogging during two natural hazards events: What twitter may contribute to situational awareness. Proceedings of the SIGCHI conference on human factors in computing systems*. ACM1079–1088.
Weng, J., & Lee, B. S. (2011). Event detection in twitter. *ICWSM, 11*, 401–408.
Wing, B. P., & Baldridge, J. (2011). *Simple supervised document geolocation with geodesic grids. Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1Association for Computational Linguistics*955–964.
Zhai, C., & Lafferty, J. (2001). *Model-based feedback in the language modeling approach to information retrieval. Proceedings of the tenth international conference on information and knowledge management*. ACM403–410.