Detecting and Disambiguating Locations Mentioned in Twitter Messages

Diana Inkpen[1], Ji Liu[1], Atefeh Farzindar[2], Farzaneh Kazemi[2], and Diman Ghazi[2]

[1] School of Electrical Engineering and Computer Science
University of Ottawa, Ottawa, ON, Canada
Diana.Inkpen@uottawa.ca
[2] NLP Technologies Inc., Montreal, QC, Canada
farzindar@nlptechnologies.ca

Abstract. Detecting the location entities mentioned in Twitter messages is useful in text mining for business, marketing or defence applications. Therefore, techniques for extracting the location entities from the Twitter textual content are needed. In this work, we approach this task in a similar manner to the Named Entity Recognition (NER) task focused only on locations, but we address a deeper task: classifying the detected locations into names of cities, provinces/states, and countries. We approach the task in a novel way, consisting in two stages. In the first stage, we train Conditional Random Fields (CRF) models with various sets of features; we collected and annotated our own dataset or training and testing. In the second stage, we resolve cases when there exist more than one place with the same name. We propose a set of heuristics for choosing the correct physical location in these cases. We report good evaluation results for both tasks.

1 Introduction

A system that automatically detects location entities from tweets can enable downstream commercial or not-for-profit applications. For example, automatic detection of event locations for individuals or group of individuals with common interests is important for marketing purposes, and also for detecting potential threats to public safety.

The extraction of location entities is not a trivial task; we cannot simply apply keyword matching due to two levels of ambiguities defined by [1]: *geo/non-geo ambiguity* and *geo/geo ambiguity*. Geo/non-geo ambiguities happen when a location entity is also a proper name (e.g., *Roberta* is a given name and the name of a city in *Georgia, United States*) or has a non-geographic meaning (e.g., *None* is a city in Italy in addition to the word *none* when lower case is ignored or when it appears at the beginning of a sentence). A geo/geo ambiguity occurs when several distinct places have the same name, as in *London, UK; London, ON, Canada; London, OH, USA; London, TX, USA; London, CA, USA*, and a few more in the

USA and other countries. Another example is the country name *China* being the name of cities in the United States and in Mexico.

As a consequence of the ambiguities, an intelligent system smarter than simple keyword matching is required. Specifically, we propose to address the geo/non-geo ambiguities by defining a named entity recognition task which focuses on locations and ignores other types of named entities. We train CRF classifiers for specific types of locations, and we experiment with several types of features, in order to choose the most appropriate ones. To deal with geo/geo ambiguities, we implement several heuristic disambiguation rules, which are shown to perform reasonably well. The consequent hybrid model is novel in the social media location extraction domain. Our contribution consists in the specific way of framing the problem in the two stages: the extraction of expressions composed of one or more words that denote locations, followed by the disambiguation to a specific physical location. Another contribution is an annotated dataset that we made available to other researchers. The fully-annotated dataset and the source code can be obtained through this link[3].

2 Related Work

Before the social media era, researchers focused on extracting locations from online contents such as news and blogs. [6] named this type of work *location normalization*. Their approach used a Maximum-entropy Markov model (MEMM) to find locations and a set of rules to disambiguate them. Their system is reported to have an overall precision of 93.8% on several news report datasets. [1] tried to associate each location mention in web pages with the place it refers to; they implemented a score-based approach to address both geo/non-geo and geo/geo ambiguities. Specifically, lexical evidences supporting the likelihood of a candidate location increases its score. When applied to Internet contents, their algorithm had an accuracy of 81.7%. [14] also focused on web pages; they assigned a weighted probability to each candidate of a location mentioned in a web page; they took into account the other locations in the same web page and the structural relations between them. [12] assumed that the true reference of a location is decided by its location prior (e.g., Paris is more likely the capital of France) and context prior (e.g., Washington is more likely the capital of USA if it has ”Wizards” in its con-

[3] `https://github.com/rex911/locdet`

text); they developed a ranking algorithm to find the most likely location reference based on the two priors, which achieved a precision of 61.34%.

Social media text (especially tweets), is very different from traditional text, since it usually contains misspellings, slangs and is short in terms of length. Consequently, detecting locations from social media texts is more challenging. [2] looked at how to exploit information about location from French tweets related to medical issues. The locations were detected by gazetteer lookup and pattern matching to map them to physical locations using a hierarchy of countries, states/provinces and cities. In case of ambiguous names, they did not fully disambiguate, but relied on users' time zones. They focused on the locations in user's profile, rather than the locations in the text of tweets. [11] detected place names in texts in a multi-lingual setting, and disambiguated them in order to visualize them on the map.

Statistical techniques were used to resolve ambiguities. For example, [10] identified the locations referenced in tweets by training a simple log-near model with just 2 features for geo/non-geo ambiguity and geo/geo ambiguity; the model achieved a precision of 15.8%. [5] identified location mentions in tweets about disasters for GIS applications; they applied off-the-shelf software, namely, the Stanford NER software to this task and compared the results to gold standards. [7] also showed that off-the-shelf NER systems achieve poor results on detecting location expressions.

3 Dataset

Annotated data are required in order to train our supervised learning system. Our work is a special case of the Named Entity Recognition task, with text being tweets and target Named Entities being specific kinds of locations. To our knowledge, a corresponding corpus does not yet exist.[4]

3.1 Data Collection

We used the Twitter API[5] to collect our own dataset. Our search queries were limited to six major cell phone brands, namely iPhone, Android, Blackberry, Windows Phone, HTC and Samsung. Twitter API allows its users to filter tweets based on their languages, geographic origins, the

[4] [7] recently released a dataset of various kinds of social media data annotated with generic location expressions, but not with cities, states/provinces, and countries).

[5] https://dev.twitter.com

time they were posted, etc. We utilized such functionality to collect only tweets written in English. Their origins, however, were not constrained, i.e., we collected tweets from all over the world. We ran the crawler from June 2013 to November 2013, and eventually collected a total of over 20 million tweets.

3.2 Manual Annotation

The amount of data we collected is overwhelming for manual annotation, but having annotated training data is essential for any supervised learning task for location detection. We therefore randomly selected 1000 tweets from each subset (corresponding to each cellphone brand) of the data, and obtained 6000 tweets for the manual annotation (more data would have taken too long to annotate).

We have defined annotation guidelines to facilitate the manual annotation task. [8] defined spatialML: an annotation schema for marking up references to places in natural language. Our annotation model is a submodel of spatialML. The process of manual annotation is described next.

Gazetteer Matching A gazetteer is a list of proper names such as people, organizations, and locations. Since we are interested only in locations, we only require a gazetteer of locations. We obtained such a gazetteer from GeoNames[6], which includes additional information such as populations and higher level administrative districts of each location. We also made several modifications, such as the removal of cities with populations smaller than 1000 (because otherwise the size of the gazetteer would be very large, and there are usually very few tweets in the low-populated areas) and removal of states and provinces outside the U.S. and Canada; we also allowed the matching of alternative names for locations. For instance, ATL, which is an alternative name for Atlanta, will be matched as a city.

We then used GATE's gazetteer matching module [4] to associate each entry in our data with all potential locations it refers to, if any. Note that, in this step, the only information we need from the gazetteer is the name and the type of each location. GATE has its own gazetteer, but we replaced it with the GeoNames gazetteer which serves our purpose better. The sizes of both gazetteers are listed in Table 1 [7]. In addition to a

[6] http://www.geonames.org

[7] The number of countries is larger than 200 because alternative names are counted; the same for states/provinces and cities.

larger size, the GeoNames contains information such as population, administrative division, latitude and longitude, which will be useful later in Section 5.

Gazetteer	Number of countries	Number of states and provinces	Number of cities
GATE	465	1215	1989
GeoNames	756	129	163285

Table 1: The sizes of the gazetteers.

Manual Filtering The first step is merely a coarse matching mechanism without any effort made to disambiguate candidate locations. E.g., the word *Georgia* would be matched to both the state of Georgia and the country in Europe.

In the next phase, we arranged for two annotators, who are graduate students with adequate knowledge of geography, to go through every entry matched to at least one of locations in the gazetteer list. The annotators are required to identify, first, whether this entry is a location; and second, what type of location this entry is. In addition, they are also asked to mark all entities that are location entities, but not detected by GATE due to misspelling, all capital letters, all small letters, or other causes. Ultimately, from the 6000 tweets, we obtained 1270 countries, 772 states or provinces, and 2327 cities.

We split the dataset so that each annotator was assigned one fraction. In addition, both annotators annotated one subset of the data containing 1000 tweets, corresponding to the search query of Android phone, in order to compute an inter-annotator agreement, which turned out to be 88%. The agreement by chance is very low, since any span of text could be marked, therefore the kappa coefficient that compensates for chance agreement is close to 0.88. The agreement between the manual annotations and those of the initial GATE gazetteer matcher in the previous step was 0.56 and 0.47, respectively for each annotator.

Annotation of True Locations Up to this point, we have identified locations and their types, i.e., geo/non-geo ambiguities are resolved, but geo/geo ambiguities still exist. For example, we have annotated the token *Toronto* as a city, but it is not clear whether it refers to *Toronto, Ontario, Canada*

or *Toronto, Ohio, USA*. Therefore we randomly choose 300 tweets from the dataset of 6000 tweets and further manually annotated the locations detected in these 300 tweets with their actual location. The actual location is denoted by a numerical ID as the value of an attribute named *trueLoc* within the XML tag. An example of annotated tweet is displayed in Table 2.

Mon Jun 24 23:52:31 +0000 2013 **<location locType='city', trueLoc='22321'>**Seguin **</location>** **<location locType='SP', trueLoc='12'>**Tx **</location>** RT @himawari0127i: #RETWEET#TEAMFAIRYROSE #TMW #TFBJP #500aday #ANDROID #JP #FF #Yes #No #RT #ipadgames #TAF #NEW #TRU #TLA #THF 51

Table 2: An example of annotation with the true location.

4 Location Entity Detection

We looked into methods designed for sequential data, because the nature of our problem is sequential. The different parts of a location such as country, state/province and city in a tweet are related and often given in a sequential order, so it seems appropriate to use sequential learning methods to automatically learn the relations between these parts of locations. We decided to use CRF as our main machine learning algorithm, because it achieved good results in similar information extraction tasks.

4.1 Designing Features

Features that are good representations of the data are important to the performance of a machine learning task. The features that we design for detecting locations are listed below:

– Bag-of-Words: To start with, we defined a sparse binary feature vector to represent each training case, i.e., each token in a sequence of tokens; all values of the feature vector are equal to 0 except one value corresponding to this token is set to 1. This feature representation is often referred to as *Bag-of-Words* or unigram features. We will use *Bag-of-Words Features* or *BOW features* to denote them, and the performance of the classifier that uses these features can be considered as the baseline in this work.

– Part-of-Speech: The intuition for incorporating Part-of-Speech tags in a location detection task is straightforward: a location can only be a noun or a proper noun. Similarly, we define a binary feature vector, where the value of each element indicates the activation of the corresponding POS tag. We later on denote these features by *POS features*.
– Left/right: Another possible indicator of whether a token is a location is its adjacent tokens and POS tags. The intuitive justification for this features is that locations in text tend to have other locations as neighbours, i.e., *Los Angeles, California, USA*; and that locations in text tend to follow prepositions, as in the phrases *live in Chicago*, *University of Toronto*. To make use of information like that, we defined another set of features that represent the tokens on the left and right side of the target token and their corresponding POS tags. These features are similar to Bag-of-Words and POS features, but instead of representing the token itself they represent the adjacent tokens. These features are later on denoted by *Window features* or *WIN features*.
– Gazetteer: Finally, a token that appears in the gazetteer is not necessarily a location; by comparison, a token that is truly a location must match one of the entries in the gazetteer. Thus, we define another binary feature which indicates whether a token is in the gazetteer. This feature is denoted by Gazetteer feature or GAZ feature in the succeeding sections.

In order to obtain BOW features and POS features, we preprocessed the dataset by tokenizing and POS tagging all the tweets. This step was done using the Twitter NLP and Part-of-Speech Tagging tool [9].

For experimental purposes, we would like to find out the impact each set of features has on the performance of the model. Therefore, we test different combinations of features and compare the accuracies of resulting models.

4.2 Experiments

Evaluation Metrics We report precision, recall and F-measure for the extracted location mentions, at both the token and the span level, to evaluate the overall performance of the trained classifiers. A token is a unit of tokenized text, usually a word; a span is a sequence of consecutive tokens. The evaluation at the span level is stricter.

Experimental Configurations In our experiments, one classifier is trained and tested for each of the location labels city, SP, and country. For the

learning process, we need to separate training and testing sets. We report results for 10-fold cross-validation, because a conventional choice for n is 10. In addition, we report results for separate training and test data (we chose 70% for training and 30% for testing). Because the data collection took several months, it is likely that we have both new and old tweets in the dataset; therefore we performed a random permutation before splitting the dataset for training and testing.

We would like to find out the contribution of each set of features in Section 4.1 to the performance of the model. To achieve a comprehensive comparison, we tested all possible combinations of features plus the BOW features. In addition, a baseline model which simply predicts a token or a span as a location if it matches one of the entries in the gazetteer mentioned in Section 3.2.

We implemented the models using an NLP package named MinorThird [3] that provides a CRF module [13] easy to use; the loss function is the log-likelihood and the learning algorithm is the gradient ascent. The loss function is convex and the learning algorithm converges fast.

4.3 Results

The results are listed in the following tables. Table 3 shows the results for countries, Table 4 for states/provinces and Table 5 for cities. To our knowledge, there is no previous work that extracts locations at these three levels, thus comparisons with other models are not feasible.

Features	Token			Span			Separate train-test sets	
	P	R	F	P	R	F	Token F	Span F
Baseline-Gazetteer Matching	0.26	0.64	0.37	0.26	0.63	0.37	—	—
Baseline-BOW	0.93	0.83	0.88	0.92	0.82	0.87	0.86	0.84
BOW+POS	0.93	0.84	0.88	0.91	0.83	0.87	0.84	0.85
BOW+GAZ	0.93	0.84	0.88	0.92	0.83	0.87	0.85	0.86
BOW+WIN	0.96	0.82	0.88	0.95	0.82	0.88	0.87	0.88
BOW+POS+ GAZ	0.93	0.84	0.88	0.92	0.83	0.87	0.85	0.86
BOW+WIN+ GAZ	0.95	0.85	0.90	0.95	0.85	0.89	0.90	0.90
BOW+POS+ WIN	0.95	0.82	0.88	0.95	0.82	0.88	0.90	0.90
BOW+POS+ WIN+GAZ	0.95	0.86	0.90	0.95	0.85	0.90	0.92	0.92

Table 3: Performance of the classifiers trained on different features for countries. Column 2 to column 7 show the results from 10-fold cross validation on the dataset of 6000 tweets; the last two columns show the results from random split of the dataset where 70% are the train set and 30% are the test set. (Same in Table 4 and Table 5)

Features	Token			Span			Separate train-test sets	
	P	R	F	P	R	F	Token F	Span F
Baseline-Gazetteer Matching	0.65	0.74	0.69	0.64	0.73	0.68	—	—
Baseline-BOW	0.90	0.78	0.84	0.89	0.80	0.84	0.80	0.84
BOW+POS	0.90	0.79	0.84	0.89	0.81	0.85	0.82	0.84
BOW+GAZ	0.88	0.81	0.84	0.89	0.82	0.85	0.79	0.80
BOW+WIN	0.93	0.77	0.84	0.93	0.78	0.85	0.80	0.81
BOW+POS+GAZ	0.90	0.80	0.85	0.90	0.82	0.86	0.78	0.82
BOW+WIN+GAZ	0.91	0.79	0.84	0.91	0.79	0.85	0.83	0.84
BOW+POS+WIN	0.92	0.78	0.85	0.92	0.79	0.85	0.80	0.81
BOW+POS+WIN+GAZ	0.91	0.79	0.85	0.91	0.80	0.85	0.84	0.83

Table 4: Performance of the classifiers trained on different features for SP.

Features	Token			Span			Separate train-test sets	
	P	R	F	P	R	F	Token F	Span F
Baseline-Gazetteer Matching	0.14	0.71	0.23	0.13	0.68	0.22	—	—
Baseline-BOW	0.91	0.59	0.71	0.87	0.56	0.68	0.70	0.68
BOW+POS	0.87	0.60	0.71	0.84	0.55	0.66	0.71	0.68
BOW+GAZ	0.84	0.77	0.80	0.81	0.75	0.78	0.78	0.75
BOW+WIN	0.87	0.71	0.78	0.85	0.69	0.76	0.77	0.77
BOW+POS+GAZ	0.85	0.78	0.81	0.82	0.75	0.78	0.79	0.77
BOW+WIN+GAZ	0.91	0.76	0.82	0.89	0.74	0.81	0.82	0.81
BOW+POS+WIN	0.82	0.76	0.79	0.80	0.75	0.77	0.80	0.79
BOW+POS+WIN+GAZ	0.89	0.77	0.83	0.87	0.75	0.81	0.81	0.82

Table 5: Performance of the classifiers trained on different features for cities.

4.4 Discussion

The results from Table 3, 4 and 5 show that the task of identifying cities is the most difficult, since the number of countries or states/provinces is by far smaller. In our gazetteer, there are over 160,000 cities, but only 756 countries and 129 states/provinces, as detailed in Table 1. A lager number of possible classes generally indicates a larger search space, and consequently a more difficult task. We also observe that the token level F-measure and the span level F-measure are quite similar, likely due to the fact that most location names contain only one word.

We also include the results when one part of the dataset (70%) is used as training data and the rest (30%) as test data. The results are slightly

different to that of 10-fold cross validation and tend to be lower in terms of f-measures, likely because less data are used for training. However, similar trends are observed across feature sets.

The baseline model not surprisingly produces the lowest precision, recall and f-measure; it suffers specifically from a dramatically low precision, since it will predict everything contained in the gazetteer to be a location. By comparing the performance of different combinations of features, we find out that the differences are most significant for the classification of cities, and least significant for the classification of states/provinces, which is consistent with the number of classes for these two types of locations. We also observe that the simplest features, namely BOW features, always produce the worst performance at both token level and span level in all three tasks; on the other hand, the combination of all features produces the best performance in every task, except for the prediction of states/provinces at span level. These results are not surprising.

We conducted t-tests on the results of models trained on all combinations of features listed in Table 3, 4 and 5. We found that in *SP* classification, no pair of feature combinations yields statistically significant difference. In *city* classification, using only BOW features produces significantly worse results than any other feature combinations at a 99.9% level of confidence, except BOW+POS features, while using all features produces significantly better results than any other feature combinations at a 99% level of confidence, except BOW+GAZ+WIN features. In *country* classification, the differences are less significant; where using all features and using BOW+GAZ+WIN features both yield significantly better results than 4 of 6 other feature combinations at a 95% level of confidence, while the difference between them is not significant; unlike in *city* classification, the results obtained by using only BOW features is significantly worse merely than the two best feature combinations mentioned above.

We further looked at the t-tests results of *city* classification to analyze what impact each feature set has on the final results. When adding POS features to a feature combination, the results might improve, but never statistically significantly; by contrast, they always significantly improve when GAZ features or WIN features are added. These are consistent with our previous observations.

4.5 Error Analysis

We went through the predictions made by the location entity detection model, picked some typical errors made by it, and looked into the

possible causes of these errors.

Example 1:

> Mon Jul 01 14:46:09 +0000 2013
> Seoul
> yellow cell phones family in South Korea #phone #mobile #yellow #samsung `http://t.co/lpsLgepcCW`

Example 2:

> Sun Sep 08 06:28:50 +0000 2013
> minnesnowta .
> So I think Steve Jobs' ghost saw me admiring the Samsung Galaxy 4 and now is messing with my phone. Stupid Steve Jobs. #iphone

In Example 1, the model predicted ”Korea” as a country, instead of ”South Korea”. A possible explanation is that in the training data there are several cases containing ”Korea” alone, which leads the model to favour ”Korea” over ”South Korea”.

In Example 2, the token ”minnesnowta” is quite clearly a reference to ”Minnesota”, which the model failed to predict. Despite the fact that we allow the model to recognize nicknames of locations, these nicknames come from the GeoNames gazetteer; any other nicknames will not be known to the model. On the other hand, if we treat ”minnesnowta” as a misspelled ”Minnesota”, it shows that we can resolve the issue of unknown nicknames by handling misspellings in a better way.

5 Location Disambiguation

5.1 Methods

In the previous section, we have identified the locations in Twitter messages and their types; however, the information about these locations is still ambiguous. In this section, we describe the heuristics that we use to identify the unique actual location referred to by an ambiguous location name. These heuristics rely on information about the type, geographic

hierarchy, latitude and longitude, and population of a certain location, which we obtained from the GeoNames Gazetteer. The disambiguation process is divided into 5 steps, as follows:

1. **Retrieving candidates**. A list of locations whose names are matched by the location name we intend to disambiguate are selected from the gazetteer. We call these locations candidates. After step 1, if no candidates are found, disambiguation is terminated; otherwise we continue to step 2.
2. **Type filtering**. The actual location's type must agree with the type that is tagged in the previous step where we apply the location detection model; therefore, we remove any candidates whose types differ from the tagged type from the list of candidates. E.g., if the location we wish to disambiguate is *Ontario* tagged as a city, then *Ontario* as a province of Canada is removed from the list of candidates, because its type *SP* differs from our target type. After step 2, if no candidates remain in the list, disambiguation is terminated; if there is only one candidate left, this location is returned as the actual location; otherwise we continue to step 3.
3. **Checking adjacent locations**. It is common for users to put related locations together in a hierarchical way, e.g., Los Angeles, California, USA. We check adjacent tokens of the target location name; if a candidate's geographic hierarchy matches any adjacent tokens, this candidate is added to a temporary list. After step 3, if the temporary list contains only one candidate, this candidate is returned as the actual location. Otherwise we continue to step 4 with the list of candidates reset.
4. **Checking global context**. Locations mentioned in a document are geographically correlated [6]. In this step, we first look for other tokens tagged as a location in the Twitter message; if none is found, we continue to step 5; otherwise, we disambiguate these context locations. After we obtain a list of locations from the context, we calculate the sum of their distances to a candidate location and return the candidate with minimal sum of distances.
5. **Default sense**. If none of the previous steps can decide a unique location, we return the candidate with largest population (based on the assumption that most tweets talk about large urban areas).

5.2 Experiments and Results

We ran the location disambiguation algorithm described above. In order to evaluate how each step (more specifically, step 3 and 4, since other steps are mandatory) contributes to the disambiguation accuracy, we also deactivated optional steps and compared the results.

Deactivated steps	Accuracy
None	95.5 %
Adjacent locations	93.7 %
Global context	**98.2** %
Adjacent locations + context locations	96.4 %

Table 6: Results on the subset of 300 tweets annotated with disambiguated locations.

The results of different location disambiguation configurations are displayed in Table 6, where we evaluate the performance of the model by accuracy, which is defined as the proportion of correctly disambiguated locations. By analyzing them, we can see that when going through all steps, we get an accuracy of 95.5%, while by simply making sure the type of the candidate is correct and choosing the default location with the largest population, we achieve a better accuracy. The best result is obtained by using the adjacent locations, which turns out to be 98.2% accurate. Thus we conclude that adjacent locations help disambiguation, while locations in the global context do not. Therefore the assumption made by [6] that the locations in the global context help the inference of a target location does not hold for Twitter messages, mainly due to their short nature.

5.3 Error Analysis

Similar to Section 4.5, this section presents an example of errors made by the location disambiguation model in Example 3. In this example, the disambiguation rules correctly predicted ”NYC” as ”New York City, New York, United States”; however, ”San Francisco” was predicted as ”San Francisco, Atlantida, Honduras”, which differs from the annotated ground truth. The error is caused by step 4 of the disambiguation rules that uses contextual locations for prediction; San Francisco of Honduras is 3055 kilometres away from the contextual location New York City, while San Francisco of California, which is the true location, is 4129

kilometres away. This indicates the fact that a more sophisticated way of dealing with the context in tweets is required to decide how it impacts the true locations of the detected entities.

6 Conclusion and Future Work

In this paper, we looked for location entities in tweets. We extracted different types of features for this task and did experiments to measure their usefulness. We trained CRF classifiers that were able to achieve a very good performance. We also defined disambiguation rules based on a few heuristics which turned out to work well. In addition, the data we collected and annotated is made available to other researchers to test their models and to compare with ours.

We identify two main directions of future work. First, the simple rule-based disambiguation approach does not handle issues like misspellings well, and can be replaced by a machine learning approach, although this requires more annotated training data. Second, since in the current model, we consider only states and provinces in the United States and Canada, we need to extend the model to include states, provinces, or regions in other countries as well. Lastly, deep learning models were shown to be able to learn helpful document level as well as word level representations, which can be fed into a sequential tagging model; we plan to experiment with this approach in the future.

References

1. Amitay, E., Har'El, N., Sivan, R., Soffer, A.: Web-a-Where: Geotagging Web Content. In: Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04. pp. 273–280. ACM Press, New York, New York, USA (Jul 2004), http://dl.acm.org/citation.cfm?id=1008992.1009040
2. Bouillot, F., Poncelet, P., Roche, M.: How and why exploit tweet ' s location information ? In: Jérôme Gensel, D.J., Vandenbroucke, D. (eds.) AGILE'2012 International Conference on Geographic Information Science. pp. 24–27. Avignon (2012)
3. Cohen, W.W.: Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data (2004)
4. Cunningham, H.: GATE, a general architecture for text engineering. Computers and the Humanities 36(2), 223–254 (2002)
5. Gelernter, J., Mushegian, N.: Geo-parsing messages from microtext. Transactions in GIS 15(6), 753–773 (2011)
6. Li, H., Srihari, R.K., Niu, C., Li, W.: Location normalization for information extraction. In: Proceedings of the 19th international conference on Computational linguistics. vol. 1, pp. 1–7. Association for Computational Linguistics, Morristown, NJ, USA (Aug 2002), http://dl.acm.org/citation.cfm?id=1072228.1072355

7. Liu, F., Vasardani, M., Baldwin, T.: Automatic identification of locative expressions from social media text: A comparative analysis. In: Proceedings of the 4th International Workshop on Location and the Web. pp. 9–16. LocWeb '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2663713.2664426
8. Mani, I., Hitzeman, J., Richer, J., Harris, D., Quimby, R., Wellner, B.: SpatialML: Annotation Scheme, Corpora, and Tools. In: Proceedings of the 6th international Conference on Language Resources and Evaluation (2008). p. 11 (2008), http://www.lrec-conf.org/proceedings/lrec2008/summaries/106.html
9. Owoputi, O., OConnor, B., Dyer, C., Gimpel, K., Schneider, N., Smith, N.A.: Improved part-of-speech tagging for online conversational text with word clusters. In: Proceedings of NAACL-HLT. pp. 380–390 (2013)
10. Paradesi, S.: Geotagging tweets using their content. In: Proceedings of the Twenty-Fourth International Florida. pp. 355–356 (2011), http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS11/paper/viewFile/2617/3058
11. Pouliquen, B., Kimler, M., Steinberger, R., Ignat, C., Oellinger, T., Blackler, K., Fluart, F., Zaghouani, W., Widiger, A., Forslund, A., Best, C.: Geocoding multilingual texts: Recognition, disambiguation and visualisation. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06). European Language Resources Association (ELRA) (2006), http://aclweb.org/anthology/L06-1349
12. Qin, T., Xiao, R., Fang, L., Xie, X., Zhang, L.: An efficient location extraction algorithm by leveraging web contextual information. In: proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. pp. 53–60. ACM (2010)
13. Sarawagi, S., Cohen, W.W.: Semi-markov conditional random fields for information extraction. In: NIPS. vol. 17, pp. 1185–1192 (2004)
14. Wang, C., Xie, X., Wang, L., Lu, Y., Ma, W.Y.: Detecting geographic locations from web resources. In: Proceedings of the 2005 workshop on Geographic information retrieval - GIR '05. p. 17. ACM Press, New York, New York, USA (Nov 2005), http://dl.acm.org/citation.cfm?id=1096985.1096991