Location Extraction from Social Media: Geoparsing, Location Disambiguation, and Geotagging

STUART E. MIDDLETON, University of Southampton IT Innovation Centre
GIORGOS KORDOPATIS-ZILOS, SYMEON PAPADOPOULOS,
and YIANNIS KOMPATSIARIS, Information Technologies Institute, CERTH

Location extraction, also called "toponym extraction," is a field covering geoparsing, extracting spatial representations from location mentions in text, and geotagging, assigning spatial coordinates to content items. This article evaluates five "best-of-class" location extraction algorithms. We develop a geoparsing algorithm using an OpenStreetMap database, and a geotagging algorithm using a language model constructed from social media tags and multiple gazetteers. Third-party work evaluated includes a DBpedia-based entity recognition and disambiguation approach, a named entity recognition and Geonames gazetteer approach, and a Google Geocoder API approach. We perform two quantitative benchmark evaluations, one geoparsing tweets and one geotagging Flickr posts, to compare all approaches. We also perform a qualitative evaluation recalling top N location mentions from tweets during major news events. The OpenStreetMap approach was best (F1 0.90+) for geoparsing English, and the language model approach was best (F1 0.66) for Turkish. The language model was best (F1@1km 0.49) for the geotagging evaluation. The map database was best (R@20 0.60+) in the qualitative evaluation. We report on strengths, weaknesses, and a detailed failure analysis for the approaches and suggest concrete areas for further research.

Categories and Subject Descriptors: **H.3.1 [Content Analysis and Indexing]**: Linguistic processing; **I.2.7 [Artificial Intelligence]**: Natural Language Processing—*Text analysis*

General Terms: Algorithms, Measurement, Performance, Design, Experimentation

Additional Key Words and Phrases: Location extraction, toponym extraction, information extraction, geoparsing, geocoding, geotagging, location, toponym, disambiguation, social media, benchmark

ACM Reference format:
Stuart E. Middleton, Giorgos Kordopatis-Zilos, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2018. Location Extraction from Social Media: Geoparsing, Location Disambiguation, and Geotagging. *ACM Trans. Inf. Syst.* 36, 4, Article 40 (June 2018), 27 pages.
https://doi.org/10.1145/3202662

This work is part of the research and development in the REVEAL project (grant agreement 610928) and TRIDEC project (grant agreement 258723), supported by the 7th Framework Program of the European Commission, and the InVID project (grant agreement 687786), supported by the Horizon 2020 Program of the European Commission.
Authors' addresses: S. E. Middleton, IT Innovation Centre, Gamma House, Enterprise Road, Southampton, SO16 7NS, UK; email: sem@it-innovation.soton.ac.uk; G. Kordopatis-Zilos, S. Papadopoulos, and Y. Kompatsiaris, Information Technologies Institute, Centre for Research and Technology Hellas, 6th km Charilaou-Thermi Road, P.O. Box 60361, 57001 Thermi-Thessaloniki, Greece; emails: {georgekordopatis, papadop, ikom}@iti.gr.

https://doi.org/10.1145/3202662

1 INTRODUCTION

Social media provides a real-time source of information with global spatial coverage that supports the daily activities of professionals in a wide variety of areas. Journalists are increasingly (Silverman 2014) turning to user-generated content from social media sites such as Twitter and Facebook to find eyewitness images and videos during breaking news events. Civil protection agencies are using social media (Earle et al. 2011) to create real-time crisis maps, plotting eyewitness damage assessments and cries for help, which can both focus ongoing relief activities and provide much-needed information to concerned friends and relatives of those involved. Businesses are using social media analytics (Chung 2014; Lassen et al. 2015) to assess the impact of product launches, mapping sentiment and getting feedback from customers discussing their experiences via social media. Any problem that needs to geospatially map information from social media posts will need a location extraction solution, which is typically implemented using geocoding, geoparsing, or geotagging in some form or another.

Geocoding, geoparsing, and geotagging are types of information extraction, which is itself a subset of information retrieval. Geocoding is the act of transforming a well-formed textual representation of an address into a valid spatial representation, such as a spatial coordinate or specific map reference. Geoparsing does the same for unstructured free text and involves location extraction and location disambiguation prior to the final geocoding. Geotagging assigns spatial coordinates to media content items, typically by building statistical models that, given a piece of text, can provide an estimate of the most likely location (spatial coordinate) to which the text refers.

It should be noted also that in the literature the term "location" is used interchangeably with the term "toponym," such as with "toponym disambiguation." We use the term "location" in this article to mean a place name (e.g., London) and consider it synonymous with the terms "location phrase" and "location mention." We use the term "disambiguated location" to mean an unambiguous location in the context of a geographic database of some type (e.g., London, UK, 51°30′26′′N 0°7′39′′W). We use the term "spatial coordinate" to mean a general reference to a coordinate such as a longitude/latitude point on a map. Sometimes "location identification" is used in the literature, which we consider to mean geoparsing without location disambiguation. Also, "location estimation" is sometimes used, which we consider to be geotagging to a spatial area such as a grid cell.

Social media content sometimes contains a geotag to indicate either the location where it was created or the location of the subject matter. Analysis has shown (Middleton et al. 2014) that about 1% of Twitter posts contain a geotag during events such as natural disasters. In the Yahoo! Flickr Creative Commons 100 Million (YFCC100m) dataset (Thomee et al. 2016), about 48% of posts have a geotag, although this percentage does not likely reflect the actual rate of geotagged images in the platform. Furthermore, a recent study has revealed that the number of geotagged photos in social media platforms such as Twitter and Flickr has tapered off (Tasse et al. 2017). Even in cases where a geotag is available, location extraction from text can still add value. Geotags can be many kilometers away from where the subject matter is located, such as long-range photographs of the Eiffel Tower. Moreover, the textual description of media posts can contain contextual location mentions that cannot be inferred from a geotag alone (e.g., "Obama in Washington making a speech about China").

There are today several commercial geocoding services, each based on an underlying map database, which can take well-formatted location descriptions and return map references to them. The problem with social media posts is that, unless posts originate from automated services such as news feeds or earthquake alerts, they are not well-formed text and therefore need some sort of parsing before they can be used with a geocoding service. There is also a problem with rate-limited remote geocoding services, of which the throughput in practice is much lower than the real-time volumes of posts common from social media sites such as Twitter.

Geoparsing can be used to process the types of unstructured text seen in social media posts and requires both location identification and location disambiguation. Approaches to location identification typically involve either named entity recognition (NER), usually based on linguistic properties such as part-of-speech tags, or named entity matching (NEM) based on a gazetteer, geospatial database, or tag set of known tokens associated with locations. The choice of gazetteer or database will depend on the spatial resolution the geoparsing is trying to operate within, ranging from the level of administrative regions, such as cities, down to street and building levels. Geotagging methods, however, are developed based on large corpora of geolocated social media posts, typically with a focus on popular locations, and can include vernacular names often missing from map databases, and even take advantage of nongeographical names and terms that are indicative of a certain location (festivals, local dishes, etc.).

Location disambiguation takes a set of possible location matches for a text token and selects the most likely ones based on available contextual evidence such as co-occurring mentions of other locations or post geotags. The approaches for location identification and disambiguation can often support each other, and hybrid approaches are not uncommon. Geotagging involves a combination of location identification and disambiguation, presented as a geotagging problem. Geotagging is often applied to estimate the location of an image or video, optionally using additional context beyond text labels such as semantic information extracted from image and video content. Section 2 provides a good overview of the types of geoparsing and geotagging approaches used today.

This article presents a comparative study among five "best-of-class" location extraction algorithms. Author-developed approaches are based on (1) entity matching using an OpenStreetMap (OSM) database and (2) a language model using a combination of a large social media tag dataset and multiple gazetteers. Third-party developed approaches are based on (3) DBpedia-based entity recognition and disambiguation, (4) named entity recognition and GeoNames gazetteer lookup, and (5) named entity recognition and the Google Geocoder API.

Our geoparsing quantitative benchmark experiments use a manually labeled Twitter dataset covering thousands of tweets during four major news events. We evaluate the precision and recall when extracting location mentions without disambiguation, working at resolutions down to street and building level. Our dataset includes within it labeled tweets shared with us by Carnegie Mellon University, allowing comparison of results to previously published work on the Christchurch earthquake event (Gelernter and Balaji 2013).

Our geotagging quantitative evaluation uses the standard Yahoo! Flickr Creative Commons 100 Million (YFCC100m) dataset (Thomee et al. 2016) containing millions of geotagged Flickr posts. We evaluate the precision and recall of our location extraction methods using a geotagging problem formulation, working at a $1km^2$ spatial precision.

Lastly, we perform a case-study-driven qualitative evaluation, taking more than 1 million tweets crawled from three recent news stories and ranking extracted locations by mention count. We examine the recall for each algorithm of ground-truth locations mentioned in published verified news stories at the time of each event.

The contribution of this article includes both original algorithm work and detailed evaluation on benchmark datasets. Two author-developed location extraction algorithms are presented, both of which have been extended from previously published work to include additional original features. The *map-database* algorithm has been extended from Middleton et al. (2014) and Middleton and Krivcovs (2016) to add location disambiguation heuristics, making use of textual context including location co-occurrences and parent region mentions, spatial proximity to geotags, person name filters, and token subsumption strategies. Its novelty lies in the use of the geographic shape information and metadata from OpenStreetMap to disambiguate locations and boost geoparsing precision. The *lm-tags-gazetteer* algorithm is based on a language model and has been extended from Kordopatis et al. (2015b), Kordopatis et al. (2016), and Kordopatis et al. (2017) to include a location-labeling step, which leverages multiple gazetteers to improve the quality of the location prediction model. The proposed approach resulted in a relative improvement of 4.5% in the geotagging performance (P@1km) compared to the best results reported in the previously published versions of the algorithm. Its novelty lies in the effective combination of multiple data sources (i.e., Flickr image metadata, OpenStreetMap, and GeoNames) in a single probabilistic location language model.

The results published in this article represent a valuable benchmark for other researchers to compare against. All benchmark datasets are freely available, and we encourage other researchers to benchmark their location extraction approaches on our labeled data against the results in this article. The author-developed algorithms are also available via open-source releases. The *map-database* algorithm has never been evaluated before on a geotagging problem formulation, and the *lm-tags-gazetteer* algorithm has never been evaluated before on a geoparsing problem formulation. Both algorithms have never been evaluated on our case-study-driven evaluation before. Our final contribution is a detailed analysis and discussion of the strengths and weakness of our approaches, providing insights to other researchers who might be considering developing their own location extraction approaches.

We report on related work in Section 2 including a discussion of the limitations in the state of the art. In Section 3, we describe details on the methods used by each of the algorithms evaluated in this article. Sections 4 and 5 provide evaluation results and discussion, and we conclude in Section 6 highlighting a few areas where we think further research might lead to improvements on the best-of-class approaches outlined in this article.

2 RELATED WORK

2.1 Geocoding Services

Commercial geocoding services such as the Google Geocoding API,[1] OpenStreetMap Nominatim,[2] and Bing Maps API[3] allow users to post a textual phrase and get back a likely location reference that matches it, along with a longitude and latitude spatial coordinate. These services expect well-formed text with super-regions provided for disambiguation. For example, sending the text "London, UK" to the Google Geocoder API will result in a spatial coordinate for the center of London, a bounding box for London, and some metadata such as the full set of UK administrative super-regions for London. Location disambiguation is generally weak or nonexistent in geocoding services due to a lack of available context. For example, sending the text "Winchester" to the Google Geocoder API will result in eight places called "Winchester" in the United States, Google's default locale. Only when one allows his or her browser to share its location (e.g., Southampton in the UK) will the result be further disambiguated to suggest, e.g., the city of Winchester, Hampshire, in the United Kingdom, which is closer to the browser location and therefore more likely to be right.

Commercial geocoding services are also subject to rate limits that prevent them from scaling up to handle high-throughput applications, such as processing real-time social media streams. At the time of this writing, the Google Geocoder API allows 2,500 free requests per day, with 100,000 allowed for premium users, and 50 requests per second. Typical breaking news events

[1] https://developers.google.com/maps/documentation/geocoding.
[2] http://wiki.openstreetmap.org/wiki/Nominatim.
[3] https://www.microsoft.com/maps/choose-your-bing-maps-API.aspx.

generate much higher volumes of content than these limits can handle. For example, recent Twitter crawls (Wiegand and Middleton 2016) of the November 2015 Paris shootings event captured about 6,000,000 posts during the first 6 hours of the event. Geocoding datasets of this size in real time with rate-limited geocoding services is not feasible, even if one was able to extract well-formatted location references from each tweet's text.

2.2 Geoparsing

Geoparsing from free text is a well-studied field in information retrieval, with applications including automated image tagging, web page annotation, and social media analytics. Approaches can be broadly categorized as either named entity recognition or named entity matching. Named entity recognition approaches are usually based on linguistic annotations, such as part-of-speech (POS) tags or bag-of-word feature vectors, which are then either used as training data for building supervised classifiers or input to hand-crafted linguistic grammars to classify and extract location mentions. Named entity matching uses a lookup index of known location names and variants, either from a gazetteer such as GeoNames[4] or from a geospatial database such as OpenStreetMap,[5] to identify possible location mentions along with heuristics to reduce false positives (FPs). These approaches can be combined, with gazetteer-based entity matching used to geocode an entity recognition result.

The earliest approaches to location entity recognition identified event locations from large text documents by analyzing co-occurrence of dates and noun phrase patterns (Swan and Allan 1999). More recently, named entity recognition of locations in social media messages has been specifically addressed. This is a challenging area due to the short text length and wide variety of grammatical styles in social media posts (Bontcheva et al. 2013). Typical approaches include conditional random fields (CRFs) coupled with named entity recognition (Ritter et al. 2011) and entity disambiguation using a reference corpus such as DBpedia (van Erp et al. 2013). Regression trees (Cheng et al. 2010) have been trained on tweet datasets with a combination of stemming and stop word removal. A best-of-class entity recognition approach is the one presented in Gelernter and Balaji (2013), which employs POS tagging, named entity recognition, a set of global and local gazetteers, and some heuristics such as spell checking and acronym processing. Our article reports results from a benchmark evaluation using the same Christchurch earthquake tweet dataset that Gelernter and Balaji (2013) used, allowing direct comparison of our results to this previous work.

For named entity matching, various gazetteers are reported as being used in the literature. The GeoNames gazetteer is perhaps the most popular choice, typically combined with heuristics such as person name filters (Gelernter et al. 2013) or demographic filters (Tahsin et al. 2016) to bias location selection to the largest area, along with DBpedia. Use of full map databases such as OpenStreetMap is possible (Middleton et al. 2014) using a location name index created from a combination of the planet-wide OpenStreetMap database and multilingual heuristics for token expansion of place types (e.g., '… street' expands to '… street' and '… st.'). This OpenStreetMap named entity matching approach (Middleton et al. 2014) is one of the approaches tested in our benchmarks and represents a best-of-class named entity matching approach.

2.3 Location Disambiguation

Early work (Smith and Crane 2001) on location disambiguation, also called toponym disambiguation, used heuristics to prune false matches and disambiguate possible location choices. For example, common person names, person titles such as "Mr." and generic place types such as "river" were

[4] http://www.geonames.org.
[5] https://www.openstreetmap.org.

used to remove false positives. For location disambiguation, contextual super-region mentions and the proximity of locations to a centroid computed from confident location matches was used.

More recently, various additional types of heuristics and contextual data have been used to disambiguate locations. This includes co-occurring and super-region mentions, place types, text capitalization, demographic data, and semantic context where available (Want et al. 2010). A geo-referenced version of WordNet has been used (Buscaldi and Rosso 2008) to calculate a conceptual density function for disambiguation. Other approaches include entity disambiguation with machine-learning techniques such as Expectation-Maximization (Davis et al. 2012) and Random Forest (Lee at al. 2015; Rafiei and Rafiei 2016). The GNIS[6] gazetteer has also been used (Amitay et al. 2004) for entity matching, exploiting contextual mentions of super-regions and an aggregated location centroid for location disambiguation.

Disambiguation of local place names can be particularly challenging. A recent analysis (Cai and Tian 2016) of local place names within a US city showed that about 17% of locations were either vernacular names or vague and hard to disambiguate. The difficult problem of geotagging vernacular place names has been examined (Pasley et al. 2007) with only limited success.

In support of location disambiguation, there is also the field of geosemantics (Lieberman and Goad 2008), which looks at contextual information relating to geoparsed location mentions, such as position modifiers or time references, for subsequent position refinement. Techniques such as spatial role learning (Kordjamshidi et al. 2012; Bastianelli et al. 2013), where phrases for position modifiers and landmarks are extracted, are helpful when references to a location include context such as "2 miles north of New York." Temporal extraction (Verhagen et al. 2010) can help to disambiguate location mentions if an event context is known.

2.4 Geotagging

Geotagging has a similar aim to geoparsing, but only a spatial point reference is sought without the need to parse an explicit reference to a known location. This has led to the development of various probabilistic and machine-learning approaches based on spatial grids that hold text statistics for different regions of the world. Geodesic grids have been computed from Wikipedia pages (Wing and Baldridge 2011) to train a naïve Bayes classifier, and for tweets (Paraskevopoulos and Palpanas 2016), where city grids are used in combination with a TF-IDF statistical measure.

Additional features such as time zones (Mahmud et al. 2014) or friend locations (Compton et al. 2014) can be used to geotag the likely home city of Twitter profiles. The spatial proximity of locations in documents can be used to statistically disambiguate geotagging results, such as in multilingual travel guidebooks (Moncla et al. 2014). Spatial proximity can also be used for supervised classifiers, such as Awamura (2015), where a support vector machine (SVM) is trained on a combination of bag-of-words features, spatial proximity, and temporal features extracted from the tweets.

For street- and building-level geotagging, language models have been tried, such as Flatow et al. (2015), where grids are learned from datasets of tweets. The approach of Serdyukov et al. (2009) records location mentions in a graph structure, encoding both spatial and semantic relations, and models the probability of a location given the presence of a tag. A boosting coefficient is applied if the tag matches a location in the GeoNames gazetteer, allowing tags for popular place and landmark names to be weighted higher. Lastly, spatially aware term selection for geotagging has been examined (Laere et al. 2014), comparing techniques such as kernel density estimation and use of Ripley's K statistic to smooth spatial occurrences of tags in Flickr posts and Wikipedia articles and showing significant performance gains for subsequent geotagging.

[6]https://nhd.usgs.gov/gnis.html.

Standard benchmark datasets exist with labeled web and/or social media posts suitable for evaluating geotagging. For example, the YFCC100m dataset (Thomee et al. 2016) contains 100 million public Flickr photos/videos, many of which have a geotag. The MediaEval workshop runs a regular Placing Task series (Choi et al. 2014) with challenges for researchers to try out algorithms on this dataset. We make use of the YFCC100m dataset as one of our benchmarks, running an updated version of the winning MediaEval 2016 placing task algorithm (Kordopatis et al. 2016) to represent a best-of-class geotagging approach. This use of a language model and gazetteer is similar to Serdyukov et al. (2009), but our approach includes a number of feature selection and weighting steps that ultimately lead to considerable gains in terms of geotagging accuracy (Kordopatis et al. 2017).

2.5 Limitations in the State of the Art

We have found that the approaches we reviewed vary considerably in terms of the real-time throughput that they can support. Using a commercial geocoding service is not scalable for third parties, since rate-limited access imposes severe restrictions on throughput even for paying customers. To do location extraction for typical real-time Twitter loads (i.e., sampled search API throughput or even firehose throughput), algorithms need to process posts in parallel. Only a few approaches (e.g., map-database) have reported results from practical experiments parallelizing their work. From the authors' own experience with Apache Storm deployments of geoparsing services, the bottlenecks for parallelization are POS tagging, named entity recognition, and (depending on implementation) language and topic model execution. For real-world social media post throughput, more than 10 nodes are required on a cluster deployment to overcome these bottlenecks with brute-force parallelization. Entity matching can be very efficient if good indexing techniques are used.

A consistent theme from the literature is that street and building name extraction performs worse than region name extraction. The reason for this is that street names, and especially building names, are not as unique as region names. This means it is much more important to exploit contextual clues when making an extraction and disambiguation decision. Region names are relatively unique and therefore much easier to identify. We consider that the improved use of contextual clues and a deeper understanding of the linguistic context where they are used is a key area for improvement over the current state of the art today.

Older approaches tended to work with a single dataset and were unable to handle the rich variety of abbreviations and vernacular names that exist in many locations of the world. Recently, there is a clear trend in the literature toward using multiple interconnected data sources (e.g., social media tags and gazetteers). We believe that this trend is likely to continue and can foresee progress being made in approaches that exploit new information sources such as personal mobile devices and connected data from the Internet of Things (IoT). The more context that is available to associate with a user's post, the better the chance of getting location extraction and disambiguation correct. The authors' language model approach in particular is helping to progress the state of the art in this direction.

3 METHODOLOGY

In this section, we describe our named entity matching algorithms using OpenStreetMap (*map-database*) and a language model approach using a combination of Flickr social media tags and various gazetteer resources (*lm-gazetteer, lm-tags, lm-tags-gazetteer*). We benchmark our two novel approaches against three other standard approaches: geocoding (*google-geocoder*), entity recognition (*ner-gazetteer*), and entity extraction (*linked-data*). The *map-database* approach described here is a substantial improvement on the original work published in Middleton et al. (2014) with the addition

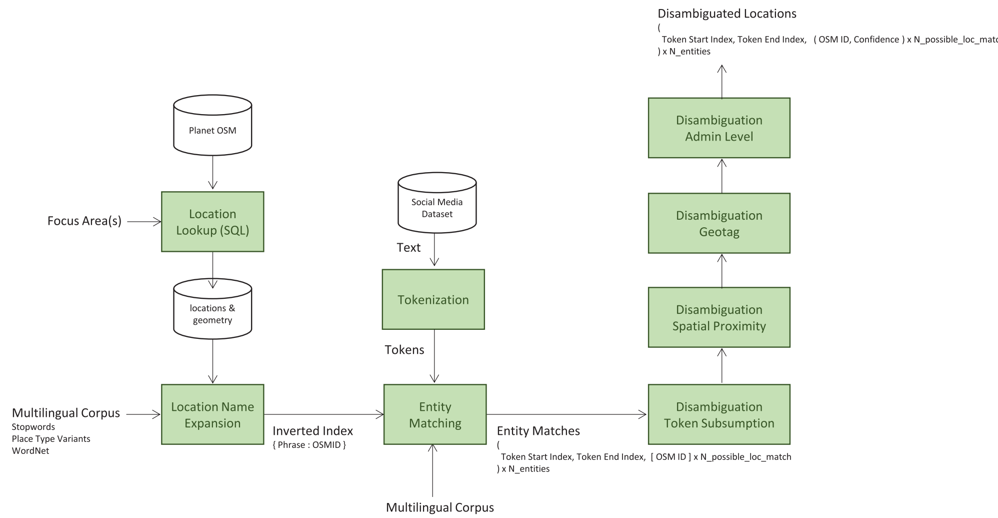

Fig. 1. Information flow pipeline for named entity matching using OpenStreetMap (map-database). In our evaluation, geotag context was not provided for disambiguation since geotags were used as ground truth.

of location disambiguation strategies. The *lm-tags-gazetteer* approach is an extension of the geotagging approach first published in Kordopatis et al. (2015b) and further analyzed in Kordopatis et al. (2017), and now includes a location-labeling step to fuse tag-based location predictions with gazetteer resources. The *google-geocoder* approach uses named entity extraction to identify possible locations in text and then sends it to the Google Geocoder API to get a location reference. The *linked-data* approach uses the linked data DBpedia spotlight service[7] (Daiber et al. 2013) to extract location entities. Finally, the *ner-gazetteer* approach uses GeoLocator[8] (Zhang and Gelernter 2014), which is based on entity recognition and entity matching from a GeoNames gazetteer.

3.1 Named Entity Matching Using OpenStreetMap: *Map-Database*

We have developed an entity matching approach that exploits region, street, and building data found in the planet OpenStreetMap database. The overall information flow for this approach can be seen in Figure 1. In an offline step, we preprocess all the location geometry held within the 400+GB planet OpenStreetMap PostGIS PostgreSQL database, generating a much smaller and more efficient database containing global region data and local street and building data for explicitly preprocessed focus areas of interest (e.g., all streets and buildings in the city of Southampton). The online geoparsing algorithm loads to an in-memory cache location names and geometry and performs entity matching in real time. Entity matches are disambiguated using available linguistic and geospatial context prior to ranking in order of confidence. The *map-database* software is packaged as *geoparsepy*[9] and is freely available from the Python Package Index (PyPi).

The preprocessing of OpenStreetMap locations involves several steps. First, a *location lookup* is performed via a PostgreSQL PostGIS query to get a set of locations to be preprocessed. A spatial filter can be applied to limit the location names that are preprocessed. The location lookup can be spatially wide (e.g., any administrative region in the world) or more focused (e.g., all streets and

[7] https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki.
[8] https://github.com/geoparser/geolocator.
[9] https://pypi.python.org/pypi/geoparsepy.

buildings in a city). Location names and the associated geometry are returned, along with references to any geometrically overlapping super-regions. The resulting smaller SQL table contains rows for each location, with columns for OSM ID, all OSM name variants including abbreviations and multilingual translations, super-region IDs, and the location polygon/line/point geometry.

The choice of focus area for location lookup and spatial filter is based on prior knowledge about which locations will be relevant. If there is no prior knowledge, then no spatial filter is selected, and a default location lookup of all global locations with an OpenStreetMap admin level of "city" or larger is used; this provides more than 300,000 location names. If a focus area is known (e.g., Christchurch, New Zealand), then all streets and buildings in the focus area are loaded in addition to the global lookup. The focus area can also be used to specify the spatial filter to remove irrelevant matches (e.g., a New Zealand spatial filter will prevent irrelevant matches about locations in China). Choosing a good location lookup will allow matching of relevant streets and buildings, whereas choosing a good spatial filter will remove irrelevant matches and false positives. In the experiments we report in this article, we used location lookups to load street and building names but did not apply any spatial filter since all global location names were considered viable matches.

The smaller preprocessed PostgreSQL PostGIS table is loaded into memory by the online geoparse algorithm on startup. A set of heuristics are applied to each OSM location name to perform *location name expansion* and create a set of n-gram location phrases. A multilingual corpus of street and building types, based on OSM feature types, is then used to compute obvious variations for common location types (e.g., "Southampton University" and "Southampton Uni"). Unigram location names that are nonnouns usually result in false positives, so they are filtered using a multilingual WordNet corpus lookup (e.g., "ok" abbreviation for Oklahoma is also used as a common adjective, "it's ok"). Location phrases are filtered using a multilingual stop word corpus and heuristics are applied to help detect OpenStreetMap labeling mistakes that occasionally appear. Once all location n-gram phrase variants are computed, an inverted index is generated so phrases can be looked up quickly and the OSM ID and geometry retrieved. A typical location cache for global regions will contain 300,000+ locations and need 8+GB of RAM. Multilingual support is provided based on corpora for English, French, German, Italian, Portuguese, Russian, and Ukrainian.

For online geoparsing, text is first cleaned and tokenized. We use the Natural Language Processing Toolkit's (NLTK's) (Bird at al. 2009) Punkt sentence tokenizer and Treebank word tokenizer. For *entity matching*, all possible n-gram tokens in a sentence are matched using the inverted location phrase cache. A corpus of common person names is used to perform a prefix check, avoiding false positives where a valid location name is actually part of a full name (e.g., Victoria Derbyshire != Derbyshire). The final result is, for each location phrase, a set of possible location matches ready for disambiguation. Disambiguation is an important step since a location name like "London" will get tens of matches across the globe ranging from the most likely (i.e., London, UK) to the pretty unlikely (i.e., London, Rusk County, Texas, US).

Location disambiguation is based on the accumulation of evidential features to create a confidence score. We first check for token subsumption, rejecting smaller gram phrases over larger ones (e.g., "New York" will prefer New York, US, to York, UK). Spatial proximity of other location mentions is then checked, with nearby parent regions and nearby locations adding to the confidence of any specific match (e.g., "New York in USA" will prefer New York, US, to New York, BO, Sierra Leone). If a geotag is available with a post, we prefer locations that are close by or overlapping to the geotag. Finally, a location with a higher OSM admin level is preferred to a location with a lower one (e.g., "New York" will prefer New York, US, to New York, BO, Sierra Leone). Once confidence scores are computed, the highest confidence location match is returned for each location phrase, with multiple options returned if several location matches have the same confidence value.

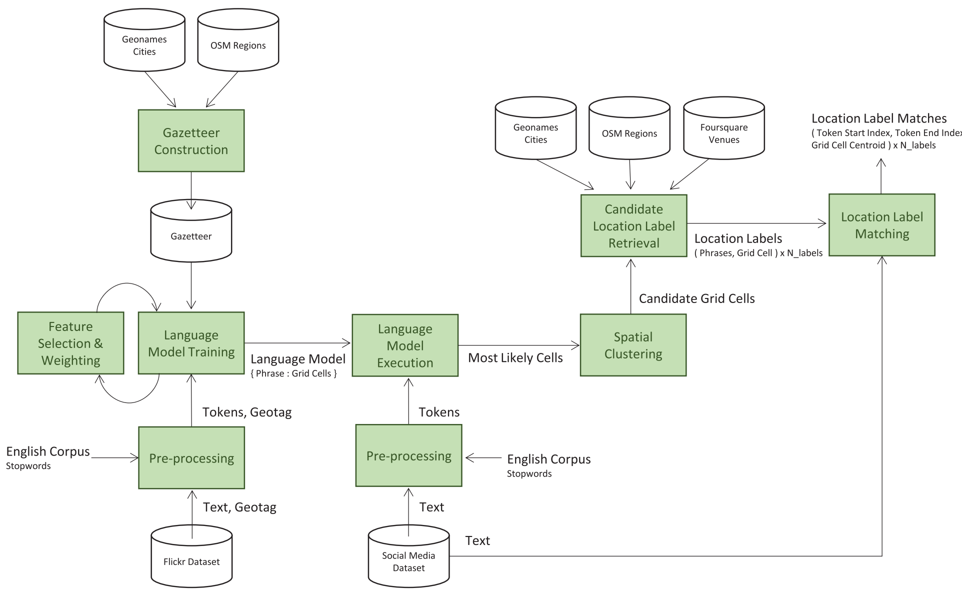

Fig. 2. Information flow pipeline for location extraction using social media tags and gazetteer (lm-tags-gazetteer).

We used the OpenStreetMap 10-point admin-level classification (e.g., country, region, city, suburb) for disambiguation in favor of demographic statistics (e.g., population size), which other researchers (Purves et al. 2007; Leidner 2008) have used. We found that in OpenStreetMap, and other gazetteers such as Geonames, the population size data is often years out of date, with locations being updated on an ad hoc basis. This means one location might have a smaller population size than another just because the two figures are reported 10 years apart. We also believe that population density is probably more important than absolute population size when it comes to disambiguation based on the likelihood of a location being talked about. Overall, we found that the OpenStreetMap admin-level classification is always correct and easily available and provides a reliable indication of the relative importance of a location to a geographic region.

For the work in this article, we preprocessed each of the cities featured in our evaluation datasets (i.e., New York, Milan, Christchurch, Paris, Brussels, Midyat) in addition to the global administrative regions. We did not use any geotag disambiguation since the geotags were used as ground truth. When multiple location options were returned with the same confidence score, we selected a random choice.

3.2 Location Extraction Using Social Media Tags and Gazetteer: *lm-tags-gazetteer*

We have extended the winning method of the MediaEval 2016 Placing Task (Kordopatis et al. 2016) to explore the use of social media tags and gazetteer information for location label extraction. A language model is first built on a corpus of geotagged Flickr posts and/or gazetteer data by analyzing their metadata and building a term-cell spatial map. Several refinements are then applied to the language model for selection and weighting of textual terms, which are highly indicative of specific locations. Figure 2 illustrates the information flow pipeline for this approach. The source implementation is freely available.[10]

[10] https://github.com/MKLab-ITI/multimedia-geotagging.

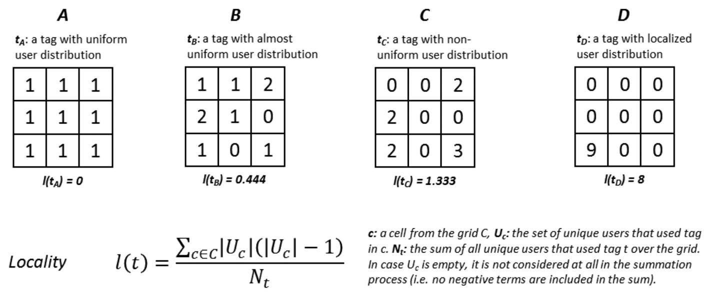

Fig. 3. Four examples of locality computation on a 3×3 toy grid, where cells represent the bounding boxes of the underlying geographic partition. In all cases, a set of nine users are assumed to have used tag t, and only their distribution to the grid cells is different.

We first apply a *preprocessing* step on the tags and titles of the dataset Flickr posts used for training. This involves URL decoding,[11] lowercase transformation, and tokenization into a term set. Multiword tag phrases are both included in their initial form (e.g., "new york") and further split into atomic tokens, which are added to the item's term set (e.g., "new," "york"). All symbols, accents, and punctuation in the terms are removed. Terms consisting only of numeric characters or less than three characters and stop words[12] are discarded. The remaining terms are used as features to build the language model as in Popescu et al. (2013).

We divide the earth's surface into (nearly) rectangular cells with a side length of 0.01° for both latitude and longitude, corresponding to a distance of approximately 1km near the equator. Then, *language model training* is performed by calculating the term-cell probabilities based on the user mentions of each term in each cell. The cell with the greater aggregate term-cell probabilities for a given query text is then considered to be the most likely cell and is used as the basis for geotagging.

After the initial construction of the language model, a *feature selection and weighting* step is applied to reduce the size of the model and increase its robustness. Two scores are extracted for each term of the language model, namely, the locality and spatial entropy scores. For feature selection only the locality score is used; however, both scores contribute to feature weighting.

Feature selection is performed based on the locality score of terms (Kordopatis et al. 2017). Locality is designed to capture the spatial distribution of term usage i.e., it quantifies how indicative a term is for a given location. It is calculated based on the number of unique users that have used the term across the spatial grid. First, for each cell c of the grid, the set of unique users U_c who made use of term t is considered. Then, locality is computed using the following equation:

$$l\,(t) = \frac{\sum_{c \in C} |U_c|\,(|U_c| - 1)}{N_t},$$

where N_t is the total number of unique users using tag t across the whole grid. Note that cells where tag t is not used at all ($|U_c| = 0$) are not considered by the summation in the nominator. Figure 3 illustrates some examples of locality computation over a 3×3 toy grid, making clear that the more uniform the tag usage distribution is, the lower the resulting locality score. All terms with a locality score of zero are discarded by the feature selection method.

The contribution of each remaining term (after feature selection) is further weighted based on its locality and spatial entropy scores. Locality weights are generated in proportion to locality scores. In particular, all terms are ranked based on their locality scores and the weights derive from the

[11]This was necessary because text in different languages was URL encoded in the YFCC100m dataset.
[12]https://sites.google.com/site/kevinbouge/stopwords-lists.

relative position of each term in the ranked term distribution. Spatial entropy weights are computed using a Gaussian weight function based on the term-cell entropy of terms (Kordopatis et al. 2015b). First, the spatial entropy of every term is calculated based on the term-cell probabilities. Then, a Gaussian function is estimated from the mean and standard deviation of the spatial entropy distribution. These weights are normalized by the maximum value in the distribution. The linear combination of the two weights is used for the generation of a weight value for every term.

To tackle the problem of extracting a location label from the text of a new social media item, the item's term set is first determined using the same *preprocessing* step as described above. Afterward, the resulting term set is fed directly to the language model to calculate the corresponding cell probabilities (language model execution).

Representing cells by their centers, a simple incremental *spatial clustering* scheme is then applied on the latitude/longitude pairs: if the ith cell is within range r from the closest one of the previous $i-1$ cells, it is assigned to its cluster; otherwise, it forms its own cluster. For every cluster, the cell with the largest probability is considered as the *cluster representative cell.* In the end, for every query item, the most likely cell and the representative cells of the clusters with more than c_t cells compose a set of "lookup cells." This set is used as the source of areas to look for geographical entities. We used $r = 2km$ and $c_t = 4$, as they were empirically found to yield the best results.

Having estimated the most likely geographic areas to contain the geographical entity of interest, we then leverage large-scale open geotagged resources, including Foursquare (FS), OpenStreetMap (OSM), and GeoNames(GN), for candidate location label retrieval. This is useful for collecting information related to local names (i.e., shop names, landmark names, etc.), addresses, and place names that fall inside the borders of the cells composing the lookup set.

We provide cell locations to the Foursquare API to get a set of nearby venues. Venues usually contain information about the name, address, and city of their respective point of interest. For every cell, five queries are sent to the Foursquare API (one for every cell corner and one for the center). For every returned venue, the values of the fields of the location of interest are stored in a gazetteer and considered as candidate location labels for the corresponding cell.

We download the complete collection of OpenStreetMap geographic areas as a gazetteer. We use only location metadata relevant to our task (i.e., names, addresses, cities/countries). Locations are organized based on the cells that they fall into. A similar process is applied to the GeoNames gazetteer where the alternative city names are used as candidate labels. For every cell in the final lookup set, a set of lowercase labels is generated from the three sources (FS, OSM, and GN).

To query the model, *location label matching* is performed that applies entity matching between the query text tokens and the label set for each lookup cell. Phrases are ranked by n-gram order, highest preferred, to help avoid partial phrase false matches.

To allow a detailed analysis of the impact that different training data have on the performance of the approach, three different language models were developed using different combinations of source data when building the language model. The different setups are (1) *tag based*, where only the tags and titles of the Flickr items contained in the YFCC100m dataset are used; (2) *gazetteer based*, where only the OpenStreetMap and GeoNames datasets are used to build the language model; and (3) *tag-gazetteer based*, where both sources are used to build the language model. The experimental results of the different setups are discussed in Section 4.

3.3 Geocoding and Named Entity Recognition Approach: *Geocoder*

To allow benchmarking against a commercial geocoding service, we developed an algorithm using simple named entity recognition and the commercial Google Geocoder API.[13] The overall information flow for this approach can be seen in Figure 4.

[13]https://developers.google.com/maps/documentation/geocoding/intro.

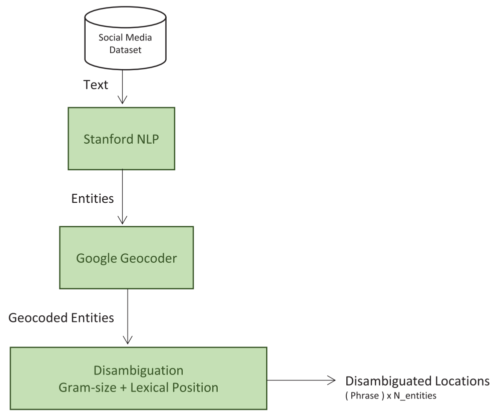

Fig. 4. Information flow pipeline for location extraction using named entity recognition and Google Geocoder API (geocoder).

The Google Geocoder API does not accept sentence input for geoparsing, only well-formatted location names. It expects well-formatted lists of location super-regions such as "Amphitheatre Parkway, Mountain View, CA." The geocoded result is a JSON-formatted list of geocoded addresses representing possible matches, including a longitude and latitude coordinate and well-formatted address with all the super-regions included. There is no confidence data associated with entries in this list, so the first result is assumed to be the best. This service is really intended to present a list of geocoded locations to a user allowing a human choice to be made as to which one is really meant by the text (e.g., for an online map search feature).

Our algorithm identifies named entities within a sentence using a combination of the Stanford POS tagger and a regex pattern that matches a sequence of proper nouns. We allow comma delimited noun phrases, so addresses with comma delimited super-regions can be passed intact to the Google Geocoder API for a better result. For each sentence, noun phrases are ordered by gram size, largest first, and lexical position, earliest first. Candidate noun phrases are sent to the Google geocoder in this order, and the first successful geocoding result used as a "best guess" disambiguated location for the entity in question. A typical sentence will have 10+ possible noun phrases to try, so there are many calls to the Google Geocoder API for every sentence. We cache geocoding results to avoid geocoding a phrase more than once, helping to reduce the number of requests to the API.

The Google Geocoder API is rate limited, with limits applying to a 24-hour period of use and per-second request rates. At the time of writing, the rate limit for the free version of the Google Geocoder API is 2,500 requests per 24 hours. We built the algorithm to pause when the rate limit was met and to wait until it was enabled again. In practice, this severely limits the size of the datasets we could geocode. For that reason, we only report in this article results using the *geocoder* algorithm for the tweet datasets and a randomly sampled 5,000-post subset from the MediaEval 2016 Placing Task Dataset; these 11,387 posts took almost 2 weeks to geocode.

3.4 Linked Data Entity Extraction Approach: *Linked-Data*

For further comparison, we have also tested an approach based on DBpedia Spotlight (Daiber et al. 2013),[14] which is a REST-based web service that exposes the functionality of annotating

[14] https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki.

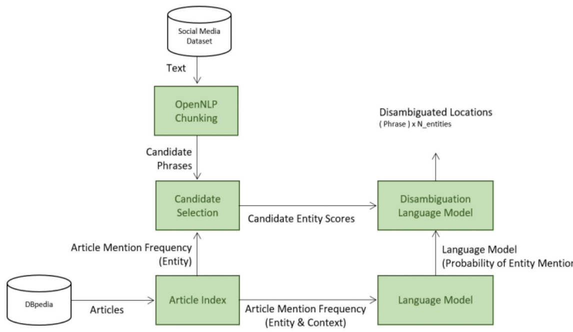

Fig. 5. Information flow pipeline for location extraction using a linked data entity extraction approach and DBpedia Spotlight service (linked-data).

and/or disambiguating DBpedia entities in text. We will refer to this approach as *linked-data.* Its information flow can be seen in Figure 5.

The DBpedia Spotlight accepts any input text and performs named entity recognition to return DBpedia URIs based on the detected entities in the input text. To do so, it uses Apache OpenNLP for phrase chunking based on noun phrase and preposition phrases, and for identifying all named entities. In addition, it selects the best candidates from the set of phrases generated from the previous step by resolving the overlap between candidates and filtering them based on a score threshold. Then, it uses a generative probabilistic language model built on Wikipedia articles, in particular based on article mention frequencies of entities. The language model produces a score for each entity given a phrase and its surrounding context, and discards entity candidates with scores lower than a certain value. The final output is the list of the resulting DBpedia entities.

To leverage DBpedia Spotlight for our problem, we limit the results returned by the service to entities that are related to places or locations. Additionally, we set a confidence threshold equal to 0.4 to ensure reliable results from the annotation process. The output of the service is a JSON-formatted list of DBpedia entities. For geoparsing tweets, their text is provided to the annotation service and a list of DBpedia entities is retrieved for each tweet. The name of each entity in the list is considered the predicted location label. Similarly, the geotagging of posts in the MediaEval 2016 Placing Task dataset is performed by giving the tags and titles as input to the DBpedia Spotlight and collecting the annotated entities. Afterward, since for this task a unique latitude/longitude pair must be estimated, we selected the location of the DBpedia entity with the largest population as the final estimate.

3.5 Named Entity Recognition and the Gazetteer Matching Approach: *Ner-gazetteer*

The last approach for comparison is the method described in Zhang and Gelernter (2014) that utilizes the geoparse algorithm from Gelernter and Zhang (2013), which we refer to as *ner-gazetteer.* We used the publicly available implementation of the method.[15] The information flow of the approach can be seen in Figure 6.

The approach receives as input either text or a tweet in JSON format and returns geographical objects that contain information about the detected locations. The method initially applies a *pre-processing* step to the input text, which involves the Stanford NLP tool and a spell checker. From this step, the input is chunked in phrases and the corresponding parts of speech are detected. The spell

[15] https://github.com/geoparser/geolocator3.

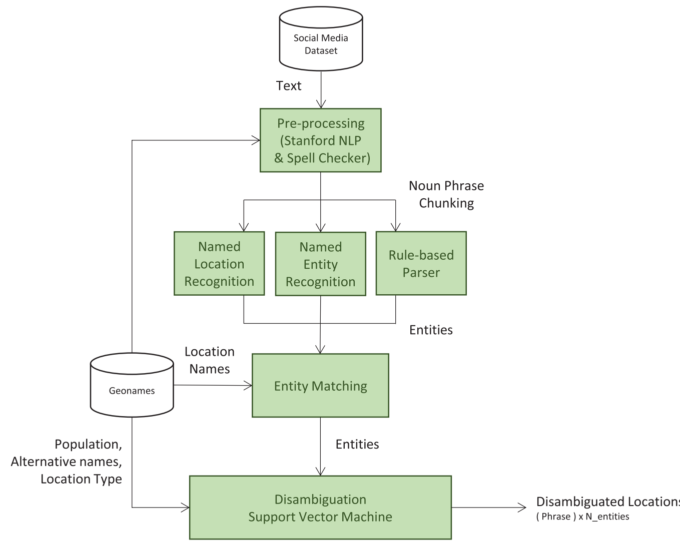

Fig. 6. Information flow pipeline for location extraction using named entity recognition and gazetteer matching approach (ner-gazetteer).

checker removes nouns that do not match with any words in a dictionary or gazetteer. The output of the preprocessing is passed to three different parsers to identify location words, i.e., the *named location parser, named entity recognition* parser, and *rule-based parser*. The recognized locations are buildings, streets, toponyms, and abbreviations. Then, *entity matching* is performed between the extracted locations and GeoNames entries to produce a set of candidate entities. The final *location disambiguation* is performed based on an SVM trained on entries with features generated based on information from GeoNames i.e., population, alternative names, and location type.

The application of this method on our data is straightforward. For geoparsing tweets, their JSON representation is directly fed to the available software implementation and the name fields of the returned locations are used as location estimates. Similar to the *linked-data* method, the tags and titles of the post contained in the MediaEval 2016 Placing Task dataset are used as input and the most populated location entity in the returned location set is considered as the estimated location of the approach for the geotagging problem.

4 EVALUATION

This section outlines the experimental analysis of the algorithms described in Section 3. We performed three different experiments and a failure analysis on the detailed results. The first experiment examined geoparsing without location disambiguation, extracting location mentions from a benchmark Twitter dataset containing a ground truth of manually labeled location mentions. The second examined geotagging to a grid cell, extracting the spatial coordinates from a large dataset of Flickr posts each associated with a ground-truth geotag. The last applied our approaches to a case study involving ranking location mentions from a Twitter feed from several real-world news events and comparing ranked location lists to ground-truth news reports.

Our experiments are designed to allow a relative performance comparison of the five considered algorithms on a variety of problem formulations and datasets. We examine each result in detail and highlight strengths and weaknesses between the algorithms. We also report on a failure analysis across all three experiments. This analysis provides the reader with a deeper discussion of which

Table 1. Breakdown of Events in Geoparse Benchmark Twitter Dataset

Event	# Tweets	Crawler Keywords	Language	Date	# Regions Mentioned	# Streets Mentioned	# Buildings Mentioned	# Locations Mentioned	Spatial mention Coverage
New York, USA Hurricane Sandy	1996	flood hurricane storm	Mostly English	Oct 2012	85	18	48	151	US South Coast
Christchurch, NZ Earthquake	2000	earthquake quake #eqnz	Mostly English	Feb 2011	33	24	64	121	New Zealand
Milan, Italy Blackout	391	blackout	Mixture Englis and Italian	May 2013	17	8	10	35	Milan
Turkey Earthquake	2000	Earthquake quake deprem	Mostly Turkish	May 2012	51	0	0	51	Turkey

algorithms work better than others on different types of text expressions and different types of location.

4.1 Geoparsing Benchmark Tweet Datasets

To examine geoparsing from text, we use the geoparse Twitter benchmark dataset (Middleton et al. 2014), available from the University of Southampton Web Observatory.[16] This open resource is available to any researcher with an interest in benchmarking geoparse algorithms, with the earliest Christchurch data evaluation reported in Gelernter and Mushegian (2013) and the earliest evaluation of the other events in this dataset reported in Middleton et al. (2014). The dataset contains a set of tweets, crawled during four news events at different times and in different countries, with manually annotated location references for mentions of administrative regions, streets, and buildings. Details for this dataset can be seen in Table 1.

It should be noted that this dataset has ground-truth labels suitable for geoparsing without location disambiguation. The human annotators were asked to label the location phrases, so they did not attempt to disambiguate locations or report spatial coordinates or map-database entries.

We ran all the algorithms on this dataset and computed location labels for each tweet. The computed location labels for each tweet were manually scored by comparing each one to the ground-truth label set provided with the dataset. If an extracted location name matched a location name in the ground-truth set, we reported a true positive (TP) for that tweet. If any extracted location name did not appear in the ground-truth set, even if another extracted location name did appear, we reported an false positive (FP) for that tweet. Tweets with no extracted labels were either a true negative (TN), if the ground-truth set was also empty, or a false negative (FN). Variant names for a location (e.g., New Zealand and NZ) were permitted as a valid match since the original ground-truth labels, created by human labelers, often used the shortest abbreviation possible for a location name. We did not differentiate between region, street, and building location granularity as we are interested in comparing geoparsing performance as a whole; Middleton et al. (2014) have previously reported results on geoparsing performance at different levels of spatial granularity.

[16]web-001.ecs.soton.ac.uk.

Precision (P) = TP/(TP + FP)	TP = true positive, FP = false positive
Recall (R) = TP/(TP + FN)	TN = true negative, FN = false negative
F1 measure = 2*PR/(P+R)	

Fig. 7. Metrics for evaluation.

The benchmark dataset ground-truth labels contain some missing location errors. This is a known issue with the dataset, resulting from the human labelers occasionally missing location mentions or not specifying the full set of location mentions in the tweet (e.g., only reporting the first location in a list of mentioned locations). We performed a meta-review of the tweets in the dataset and identified all such missing labels. There were a total of 259 meta-review tweets with a missed location in the dataset. Whenever any of our approaches correctly extracted a location that matched a meta-review missing location, we reported results as if the approach had not extracted any location at all; this resulted in either a TN or FN result. The alternative was to report a strict FP result, which would cause a misleadingly low precision score as all the meta-review missing locations are perfectly valid mentions of a location by a tweet. We followed this meta-review procedure to ensure our results can be compared directly yet fairly with the original work from Gelernter and Mushegian (2013). Our meta-review location label list is available, on request to the authors, to any researcher who needs it in the future and will be included in future releases of the geoparse Twitter benchmark dataset.

To remove any potential training bias, we filtered from the training set of Flickr posts used by the *lm-tag* and *lm-tag-gazetteer* approaches any reference to hashtags or event-specific terms for each of the four events (e.g., #EQNZ, #Milano, Sandy, Blackout). We also removed Flickr posts in the temporal period of these events.

Once all the tweets were scored, we computed precision, recall, and F1 scores. The metrics used are shown in Figure 7 and the obtained scores in Figure 8. Overall, we find that the *map-database* approach is the most robust choice for English and Italian tweets with F1 scores between 0.90 and 0.97. It provided a high precision as it was able to use context in the tweets to remove many false positives. The *geocoder* approach performed worst, due mostly to the Google geocoder matching global locations to common phrases (e.g., "deprem" which is earthquake in Turkish) and names (e.g., "Sandy"). Examples of common failure patterns are provided in Section 4.4.

The conclusion from the Turkish results is that the decision of which approach performs best appears to be sensitive to whether there is a parser available for the target language.

4.2 Geotagging Benchmark Flickr Posts

To examine geotagging, we ran each of our approaches where identified location mentions in text are returned as a spatial geotag for subsequent evaluation. For this work, we used the standard Yahoo! Flickr Creative Commons 100 Million (YFCC100m) dataset (Thomee et al. 2016), used also by the MediaEval 2016 benchmarking activity. This dataset has an in-built ground truth since each Flickr post contains a geotag. We used the same test and training set split as MediaEval 2016, with a training set size of ≈38.2M posts and test set size of ≈1.5M posts. The title, description, and tags were all available to be used for location extraction. Details for this dataset can be seen in Table 2 or explored on the dataset website.[17]

[17] http://www.yfcc100m.org/globalstats.

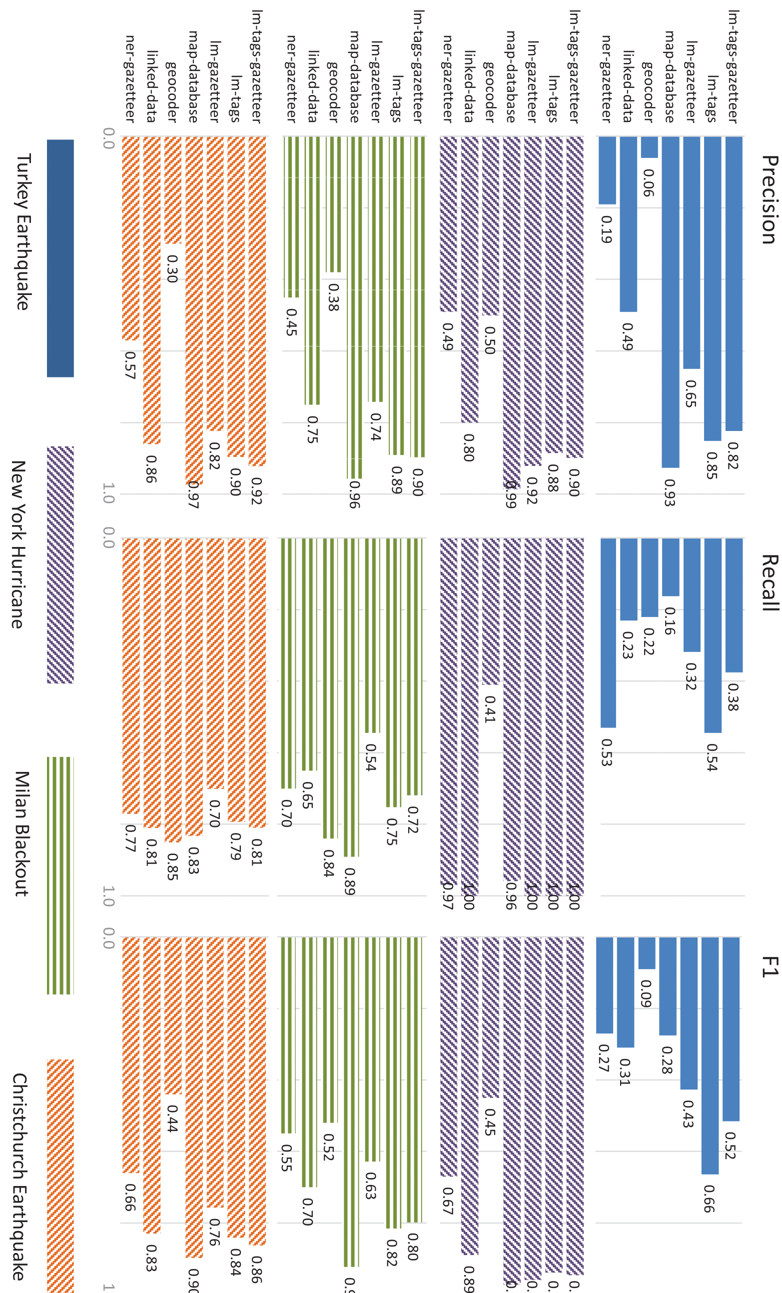

Fig. 8. Geoparsing results for benchmark Twitter posts broken down by event. The *map-database* approach had the best precision (P 0.93–0.99) overall across the four events. The *map-database* also had the best overall F1 score (F1 0.90–0.97). The *lm-tags* approach was a robust choice (F1 0.66) for the Turkey earthquake event.

Table 2. Global Statistics for YFCC100m Dataset and Our Training/Test Set Split

YFCC100m Flickr Dataset	
Timeline	February 2000–May 2014
# posts	100,000,000
# posts with 0 user tags	31,000,000
# posts with 1 user tag	7,100,000
# posts with 2 user tag	7,400,000
# posts with 3+ user tag	53,800,000
# posts-training set[18]	38,253,003
# posts-test set	1,527,398

To evaluate the results, we computed the distance between the geotag calculated by each approach and the ground-truth geotag of the Flickr post. Any geotag within a 1km radius of the ground truth was considered a TP; otherwise, it was an FP. If no geotag was returned, due to a lack of confidence in the result, it was an FN. This allowed us to compute the P@1km result and its associated recall and F1 scores.

Given the nature of Flickr, there is a potential social bias in the YFCC100m dataset where popular locations are overrepresented in terms of post frequency. It is also likely that results from the "long tail" of less frequently mentioned locations would become hidden in the mean P/R/F1 results for each grid cell. To help assess the impact of this possible bias, we computed a 1km^2 grid across the globe and assigned to each grid square a randomly chosen post of which the geotag fell within the square. We then used this reduced dataset of 286,564 posts, spatially balanced so each grid square had a single post, to compute a P@1km_square result. We call this dataset a "geographically normalized dataset."

Last, we created a smaller random sample of 5,000 posts from the full dataset to allow us to run the *geocoder* algorithm. The *geocoder* has to work within Google Geocoder rate limits and geocoding 5,000 posts takes about a week, with the algorithm pausing each day until the 24-hour rate limit is refreshed.

Results for all algorithms can be seen in Figure 9. For our algorithms, we selected confidence thresholds that optimized the F1 score. Location matches falling below the confidence threshold were ignored, usually resulting in an FN result. The *lm-tags-gazetteer* performed best overall (P@1km 0.36, F1@1km 0.49). It is clear that social tag features are dominant, since the *lm-tags* approach is almost as good as the *lm-tags-gazetteer* approach. All the approaches showed weaker results when geosocial bias was removed, using 1km^2 grid cells with equal post density, showing there is some bias in the YFCC100m dataset toward popular locations with many Flickr posts. The random sample dataset results were very similar to the full MediaEval 2016 dataset results, so the geocoder results are representative despite the small sample size.

4.3 Case Study for Location Analytics over Tweets During a Breaking News Event

We wanted to evaluate our approaches on the real-world problem of location mining from news content to see if locations with a high mention frequency, extracted from content during a breaking news event, matched the locations finally reported in news articles via respected news

[18] Although YFCC comprises 100 million posts, only approximately 40 million of them were usable for training-testing our geotagging approach. The reason for that is that approximately half of the posts do not carry any geotag information, whereas an additional 10M posts created by users belong to the test set and were excluded to avoid overfitting.

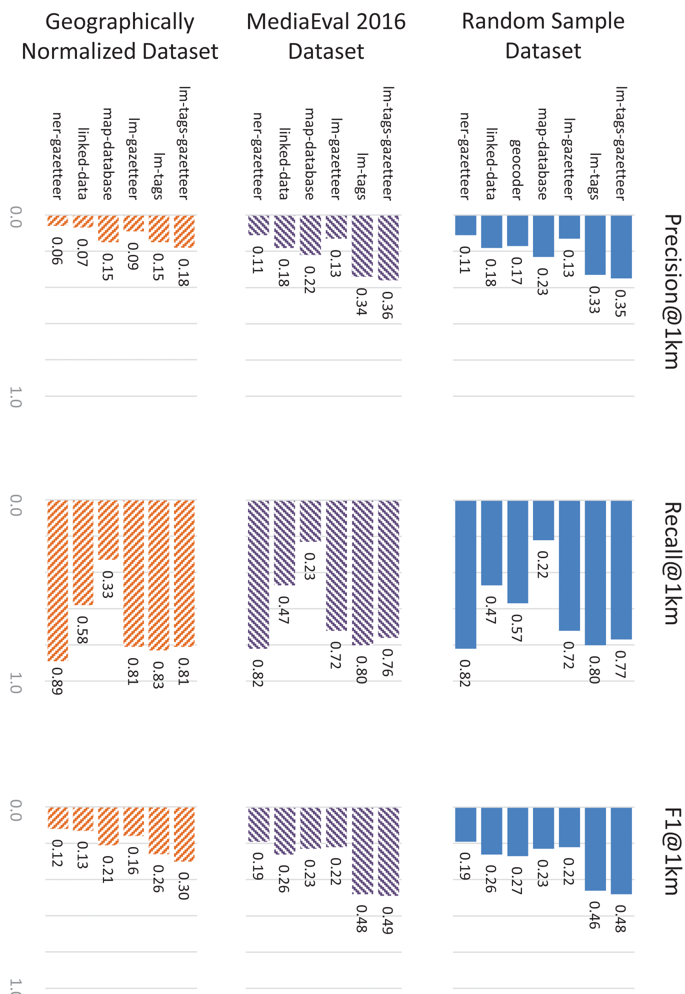

Fig. 9. Geotagging benchmark Flickr posts. The *lm-tags-gazetteer* approach was best overall (P@1km 0.36, F1@1km 0.49). The *lm-tags* approach was almost as good, showing tag features were very important. All the approaches were weaker when testing on the sampled 1km^2 grid cells, with the *lm-tags-gazetteer* approach performing best (P@1km 0.18, F1@1km 0.30). This shows there is some bias in the full dataset toward popular locations with many Flickr posts. The random sample results are very similar to full MediaEval 2016 dataset results, so the geocoder results are representative.

sites. The aim was to provide a qualitative evaluation for the recall of newsworthy locations. Journalists are under a lot of time pressure, so they are very interested in highly filtered information feeds where only pertinent data is presented. Journalists want to look at the most important locations for a breaking news story and verify the earliest posts about each incident(s) at the event location(s). Therefore, understanding the recall of newsworthy locations from a top N filtered list is an important applied use case to evaluate.

We used the Twitter search API to crawl tweets for three news events shortly after they broke. Events include the November 2015 Paris shootings, March 2016 Brussels airport bombing, and May 2016 Turkish police station bombing. Each of these events had many mainstream news reports, and

Table 3. Statistics for Twitter Datasets and Ground-Truth Locations Associated with Each News Event

Event	Crawler Keywords	# Tweets	# Ground-Truth Locations	Start Timestamp	End Timestamp
Paris shootings[19] November 13, 2015 Incidents 20:20, 20:25, 20:32, 20:40, 20:36, 20:40 UTC	paris shootings	62,908	4 (regions) 2 (streets) 5 (places)	20:20 UTC	21:20 UTC
Brussels airport bombing[20] March 22, 2016 Incidents 08:00, 09:11 UTC	brussels bombing	969,524	3 (regions) 2 (places)	04:00 UTC	10:30 UTC
Turkish police station bombing[21,22] May 8, 2016 Incident 06:00 UTC	turkey bombing midyat	57,902	3 (regions)	05:30 UTC	14:00 UTC

the journalist-verified location breakdown associated with specific incidents during each event is very well documented and is used as the ground truth for the task. The news sites used for our ground-truth location list were BBC News, CNN, and RT. The dataset statistics can be seen in Table 3.

We ran all algorithms except the *geocoder* approach, due to the aforementioned API limitations, on this dataset and geoparsed every tweet. We then compiled a ranked list of ground-truth locations for each event, ordered by frequency of mention. Finally, we computed a recall@N metric by counting the number of ground-truth locations in the top N locations extracted. The idea is to see how a top N list of trending locations extracted using the proposed methods maps to the location list used in the journalists' final news reports.

The results can be seen in Figure 10. We report R@3, R@10, R@20, and R@All as we wanted to see how coverage varied between different "top N" location sets. The *map-database* approach was best overall, with results worse when locations included streets and building names rather than just region names.

4.4 Failure Analysis

We observed some recurring patterns of location mentions that cause problems for different classes of algorithms. Table 4 shows a set of common patterns that caused problems for some algorithms.

All approaches could handle poorly formatted location mentions with the exception of Google Geocoder, which expected a well-formatted address with the primary location followed by a comma-separated list of super-regions.

The issue of spelling mistakes was addressed by some approaches by using spell checkers at the phrase extraction stage. With the exception of Google Geocoder API, which has access to a

[19] http://www.bbc.co.uk/news/world-europe-34818994.
[20] http://www.bbc.co.uk/news/world-europe-35869985.
[21] https://www.rt.com/news/345822-mardin-police-station-bomb/.
[22] http://edition.cnn.com/2016/06/08/europe/turkey-midyat-car-bomb/.

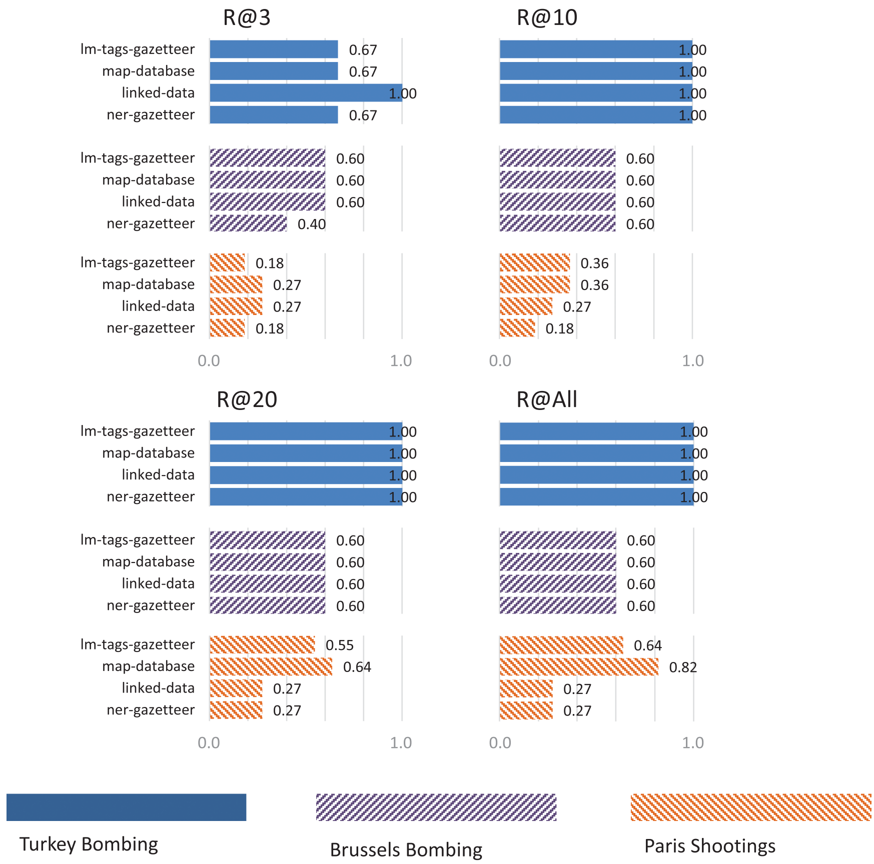

Fig. 10. Case study results ranking extracted locations mentioned in Twitter posts of breaking news stories. The *map-database* approach was best overall (R@20 0.60–1.00). Both approaches struggled with the Paris event, where the story involved streets, buildings, and regions as opposed to mostly regions.

comprehensive multilingual spell checker, the use of spell checkers imposes language restrictions (e.g., English only) as spell checking is highly language dependent. Interestingly, we found that the option of using machine translation to English and then applying spell checking was not adopted in the approaches we reviewed. This is probably due to the information loss that would occur and the fact that it is unnecessary when you have access to multilingual variant names (e.g., via OpenStreetMap) or social tags in a variety of languages.

Most approaches struggled with street and building names for unpopular locations without a social media tag. For example, news events can occur in any place, not just tourist spots, so locations of interest can be buildings or streets in unheard-of locations. The map-database and lm-tags-gazetteer were most resilient to this as they had access to detailed street and building data for focus areas (e.g., all of Christchurch's streets and buildings).

Approaches without additional vocabulary support for stop word lists, place type abbreviations, and people names consistently returned incorrect global location matches. This was mostly due to a lack of understanding of the context where the phrases were being used. Approaches using only gazetteer or linked data lists of locations were vulnerable to this. Google Geocoder was particularly weak on this point due to its strategy of returning long lists of possible location matches; the

Table 4. Examples of Failure Patterns

Pattern *(Frequency Seen from Manual Inspection)*	Algorithms That Had Trouble	Example	Correct Location
Common terms mistaken for location names *(very common)*	geocoder ner-gazetteer	This is the end of my Hurricane Sandy live-tweeting day 1	None. Mistaken location was Hurricane, UT 84737, USA
People's names that are also location names *(common)*	geocoder linked-data ner-gazetteer	Webgrrls hosting company is flooded by #Sandy	Sandy, UT, USA
Locations without any context *(common)*	geocoder ner-gazetteer	The city has high winds and flooding by the coastal lines	City of London, London, UK
Not in a well-formatted address *(rare)*	geocoder	Street flooding #NYC: 48th Ave	48th St., New York, NY, USA
Spelling mistakes *(rare)*	map-database linked-data ner-gazetteer lm-tags-gazetteer	Earthquake in ChrristchurchNew Zealand ghastly	Christchurch, New Zealand
Saints and people's title confused with place type abbreviations *(rare)*	geocoder linked-data lm-tags-gazetteer	I agree with St. Mary on this topic	None. Mistaken location was 1928 St. Marys Rd., Moraga, CA 94575, USA
Vernacular names and abbreviations *(very rare on average but depends on event)*	map-database linked-data ner-gazetteer	CHCH hospital has been evacuated	Christchurch Hospital, 2 Riccarton Ave., Christchurch Central, Christchurch 8011, New Zealand
Street names in unpopular locations *(very rare on average but depends on event)*	linked-data ner-gazetteer	Anyone have news of St. Margarets Girls College Winchester St Merivale	Margarets Girls College, 12 Winchester St., Canterbury 8014, New Zealand

intention is really for a human to choose from this list, and it provides little context to allow a high-precision automated selection.

Last, vernacular names for locations were problematic to all approaches that did not have access to social media tag data. The map databases like OpenStreetMap have some vernacular names, but often local nicknames were missing and simply ignored unless they appeared in a social media tag.

The next section reviews these results in more detail, examining possible causes of some of the observed patterns we see in the results.

5 DISCUSSION

We performed two quantitative evaluations and one qualitative evaluation to examine the different strengths and weaknesses of our approaches. Each approach had a different strength and weakness profile, allowing some interesting insights and indicating that there is potential for a complementary fusion approach to be developed.

The *map-database* was strongest for geoparsing. Locations associated with news events can be anywhere and often include places or regions that are not popularly visited or posted about on social media sites such as Flickr. We found that approaches relying on linguistic processing (e.g., part-of-speech patterns, spell checking, name filters, etc.) provided the best recall (R 0.93) and were very good at extracting location mentions from English posts. We think this is because even though social media posts have poor grammar, they contain enough linguistic clues to make effective guesses.

The *map-database* approach had a consistently high precision (P 0.93–0.99), exhibiting a strong multilingual performance with the exception of Turkish. The *lm-tags* approach, using tag set training data to list many of the popular ways people refer to a location and nonlocation contextual tags, showed its strength on the Turkish dataset (R 0.54 and up to three times better than other approaches).

When comparing how approaches performed on the problem of geotagging, the *lm-tags* approach performed best (F1 0.49). Looking into why this might be, we think that the presence of specific contextual tags in addition to location name references provided highly discriminating features, which the *lm-tags* approach could train upon. Tags such as sporting event names, food types, and local nicknames reported alongside a location name proved to be good discriminators between locations with the same name. The results degrade (F1 0.30, down 60% from 0.49) when posts are uniformly sampled on $1km^2$ grids, showing that scarcity of tags in unpopular areas does downgrade performance. This is expected as there is more training data for popular tourist locations than the more obscure locations rarely posted about. It is also true that locations where no Flickr posts exist at all are not represented in the YFCC100m dataset. However, the *lm-tags* approach clearly outperformed the others, with F1 scores double other approaches, and this shows the major strength of using tag sets for geotagging.

It would be possible to develop a hybrid approach where entity recognition is used for identification of location labels, followed by coarse-grain tag-based geotagging (e.g., get all locations within a $10km^2$ cell) and finally a fine-grain map-based entity lookup. This could be a very successful strategy where text documents are well organized, with good grammar and strong use of case for proper names, such as formal reports. In these cases, linguistic processing should detect locations with high precision. Unfortunately, social media posts are rarely well formatted, containing bad grammar and difficult nontextual content such as emojis and characters for visual expression and emphasis, so the scope for hybrid success is more limited.

For the qualitative evaluation, on our applied use case of mining breaking news tweets, we found that the *map-database* approach was the most robust (R@20 0.64–1.00). The recall errors observed with the *lm-tag-gazetteer* approach were mostly due to making a wrong location estimate in the first part of the method, resulting in a failure to find a matching location label at the next step, given that only locations from the estimated cells are considered in the matching.

It is clear from the qualitative evaluation that most techniques are weak when extracting street and building location names compared to strong performance in region name extraction. Approaches with access to a full map database, as opposed to just a gazetteer with region names, were strongest. Approaches using social media tags performed well on popular streets and buildings but completely missed mentions of place names that were not tagged (e.g., a police station in a small region of Turkey). When selecting the best method, it is therefore important to consider the use case in which location extraction will be applied. Are streets and buildings needed in addition to regions? Is the area likely to be well tagged via social media? Are focus areas known in advance? These questions need to be answered before the best technique can be recommended.

6 CONCLUSIONS

We present in this article a comprehensive analysis of five best-of-class approaches to location extraction from social media text. The first, an extension of Middleton et al. (2014), uses a location entity matching approach based on the OpenStreetMap database. The second approach, an extension of Kordopatis et al. (2017), uses a combination of training a language model on a set of Flickr post tags and a set of gazetteers. For benchmarking, we also evaluated a DBpedia-linked data matching approach, a gazetteer and named entity matching approach, and an approach based on Google geocoder lookups of named entities.

We evaluate geoparsing without location disambiguation using a standard geoparse Twitter benchmark dataset (i.e., more than 6,000 tweets), allowing us to directly compare results across all five approaches. We found that the *map-database* entity matching was best overall for English and Italian (P 0.96–0.99, F1 0.90–0.97). For Turkish, the *lm-tags* approach was best (F1 0.66).

We then evaluated geotagging by exploring a geotagging problem formulation using the YFCC100m Flickr dataset (i.e., more than 39 million geotagged Flickr posts). The *lm-tags-gazetteer* approach was strongest (F1@1km 0.49) and showed the strength of using tag sets for location disambiguation. It should be noted, however, that there are locations, especially unpopular or insignificant locations, where there are no Flickr posts at all, which would represent areas where approaches using tags alone would fail.

We lastly performed an applied qualitative evaluation where datasets of breaking news tweets (i.e., more than 1 million tweets) were geoparsed and the results, ranked by top-N mention frequency, were compared to locations published in ground-truth news reports from BBC, CNN, and RT. The *map-database* approach was strongest (R@20 0.60+), probably due to the fact that OpenStreetMap has no variation in coverage between popular and unpopular locations and was able to successfully identify lesser-known street and suburb names from the news reports.

There are a few areas where further research might improve on the best-of-class approaches outlined in this article. The first is to explore other sources of context to social media posts, such as interconnected datasets from mobile devices and the Internet of Things (IoT). If real-time geoparsing is not required, then adaptive lookup and indexing of social media sites with tagged content could be performed to add user profile context to posts. The last area is the use of location refinement strategies, exploiting available geosemantic context within the text of each post. Sometimes location mentions come with some geosemantic context such as "5 miles north of London." This could be parsed and the spatial location reference adjusted accordingly to improve geotagging precision.

ACKNOWLEDGMENTS

The authors would like to thank Judith Gelernter from Carnegie Mellon University for providing access to the labeled Christchurch, New Zealand, earthquake tweet dataset she used for her original 2011 work.

REFERENCES

Ritesh J. Agrawal and James G. Shanahan. 2010. Location disambiguation in local searches using gradient boosted decision trees. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10)*. ACM, New York, NY, 129–136. DOI: 10.1145/1869790.1869811 http://doi.acm.org/10.1145/1869790.1869811

Einat Amitay, Nadav Har'El, Ron Sivan, and Aya Soffer. 2004. Web-a-where: Geotagging web content. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04)*. ACM, New York, NY, 273–280. DOI: http://dx.doi.org/10.1145/1008992.1009040

Kalina Bontcheva, Leon Derczynski, Adam Funk, Mark A. Greenwood, Diana Maynard, and Niraj Aswani. 2013. TwitIE: An open-source information extraction pipeline for microblog text. In *Proceedings of Recent Advances in Natural Language Processing*. 83–90.

Davide Buscaldi and Paolo Rosso. 2008. A conceptual density-based approach for the disambiguation of toponyms. *International Journal of Geographical Information Systems* 22, 3 (2008), 301–313.

Guoray Cai and Ye Tian. 2016. Towards geo-referencing infrastructure for local news. In *Proceedings of the 10th Workshop on Geographic Information Retrieval (GIR'16)*. ACM, New York, Article 9, 10 pages. DOI: https://doi.org/10.1145/3003464.3003473

Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. You are where you tweet: A content-based approach to geo-locating Twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*. ACM, New York, NY, 759–768. DOI: http://dx.doi.org/10.1145/1871437.1871535

Jaeyoung Choi, Bart Thomee, Gerald Friedland, Liangliang Cao, Karl Ni, Damian Borth, Benjamin Elizalde, Luke Gottlieb, Carmen Carrano, Roger Pearce, and Doug Poland. 2014. The placing task: A large-scale geo-estimation challenge for social-media videos and images. In *Proceedings of the 3rd ACM Multimedia Workshop on Geotagging and Its Applications in Multimedia (GeoMM'14)*. ACM, New York, NY, 27–31. DOI: http://dx.doi.org/10.1145/2661118.2661125

Wingyan Chung. 2014. BizPro: Extracting and categorizing business intelligence factors from textual news articles. *International Journal of Information Management* 34, 2 (2014), 272–284.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics'13)*.

Paul Earle, Daniel C. Bowden, and Michelle Guy. 2011. Twitter earthquake detection: Earthquake monitoring in a social world. *Annals of Geophysics* 54, 6 (2011), 708–715.

David Flatow, Mor Naaman, Ke Eddie Xie, Yana Volkovich, and Yaron Kanza. 2015. On the accuracy of hyper-local geotagging of social media content. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining (WSDM'15)*. ACM, New York, NY, 127–136. DOI: http://dx.doi.org/10.1145/2684822.2685296

Judith Gelernter and Shilpa Balaji. 2013. An algorithm for local geoparsing of microtext. *Geoinformatica* 17, 4 (2013), 635–667. DOI: http://dx.doi.org/10.1007/s10707-012-0173-8

Judith Gelernter and Wei Zhang. 2013. Cross-lingual geo-parsing for non-structured data. In *Proceedings of the 7th Workshop on Geographic Information Retrieval*. ACM, New York, NY, 64–71. DOI: 10.1145/2533888.2533943

Giorgos Kordopatis-Zilos, Adrian Popescu, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2015. CERTH/CEA LIST at MediaEval Placing Task 2015. *MediaEval 2015*.

Giorgos Kordopatis-Zilos, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2015. Geotagging social media content with a refined language modelling approach. In *Proceedings of the Pacific-Asia Workshop on Intelligence and Security Informatics*.

Giorgos Kordopatis-Zilos, Adrian Popescu, Symeon Papadopoulos, and Yannis Kompatsiaris. 2016. Placing images with refined language models and similarity search with PCA-reduced VGG features. In *Proceedings of MediaEval Workshop 2016*.

Giorgos Kordopatis-Zilos, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2017. Geotagging text content with language models and feature mining. *Proceedings of the IEEE* 105, 10 (2017), 1971–1986. DOI: https://doi.org/10.1109/JPROC.2017.2688799

Olivier Van Laere, Jonathan Quinn, Steven Schockaert, and Bart Dhoedt. 2014. Spatially aware term selection for geotagging. *IEEE Transactions on Knowledge and Data Engineering* 26, 1 (2014), 221–234.

Niels Buus Lassen, Rene Madsen, and Ravi Vatrapu. 2014. Predicting iPhone sales from iPhone tweets. In *Proceedings of the 2014 IEEE 18th International Enterprise Distributed Object Computing Conference (EDOC'14)*. IEEE, Los Alamitos, CA, 81–90. DOI: http://dx.doi.org/10.1109/EDOC.2014.20

Sunshin Lee, Mohamed Farag, Tarek Kanan, and Edward A. Fox. 2015. Read between the lines: A machine learning approach for disambiguating the geo-location of tweets. In *Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'15)*. ACM, New York, NY, 273–274. DOI: http://dx.doi.org/10.1145/2756406.2756971

Jochen L. Leidner. 2008, *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names*. Ph.D. Dissertation. School of Informatics, University of Edinburgh.

Jalal Mahmud, Jeffrey Nichols, and Clemens Drews. 2014. Home location identification of Twitter users. *ACM Transactions on Intelligent Systems and Technologies* 5, 3, Article 47, 21 pages. DOI: http://dx.doi.org/10.1145/2528548

Stuart E. Middleton, Lee Middleton, and Stefano Modafferi. 2014. Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems* 29, 2 (2104), 9–17.

Stuart E. Middleton and Vadims Krivcovs. 2016. Geoparsing and geosemantics for social media: Spatio-temporal grounding of content propagating rumours to support trust and veracity analysis during breaking news. *ACM Transactions on Information Systems* 34, 3, Article 16, 26 pages. DOI: 10.1145/2842604

Ludovic Moncla, Walter Renteria-Agualimpia, Javier Nogueras-Iso, and Mauro Gaio. 2014. Geocoding for texts with fine-grain toponyms: an experiment on a geoparsed hiking descriptions corpus. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'14)*. ACM, New York, NY, 183–192. DOI: http://dx.doi.org/10.1145/2666310.2666386

Pavlos Paraskevopoulos and Themis Palpanas. 2016. Where has this tweet come from? Fast and fine-grained geolocalization of non-geotagged tweets. *Social Network Analysis and Mining* 6 (2016), 89. DOI: https://doi.org/10.1007/s13278-016-0400-7

Robert C. Pasley, Paul D. Clough, and Mark Sanderson. 2007. Geo-tagging for imprecise regions of different sizes. In *Proceedings of the 4th ACM Workshop on Geographical Information Retrieval (GIR'07)*. ACM, New York, NY, 77–82. DOI: http://dx.doi.org/10.1145/1316948.1316969

Adrian Popescu and Nicolas Ballas. 2013. CEA LIST's participation at MediaEval 2013 placing task. In *Proceedings of MediaEval Workshop 2013*.

Ross S. Purves, Paul Clough, Christopher B. Jones, Avi Arampatzis, Benedicte Bucher, David Finch, Gaihua Fu, Hideo Joho, Awase Khirni Syed, Subodh Vaid, and Bisheng Yang. 2007. The design and implementation of SPIRIT: A spatially aware search engine for information retrieval on the Internet. *International Journal of Geographical Information Science* 21, 7 (2007), 717–745. DOI: http://dx.doi.org/10.1080/13658810601169840

Compton Ryan, David Jurgens, and David Allen. 2014. Geotagging one hundred million Twitter accounts with total variation minimization. In *Proceedings of the IEEE International Conference on Big Data (Big Data'14)*.

Pavel Serdyukov, Vanessa Murdock, and Roelof van Zwol. 2009. Placing Flickr photos on a map. In *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*. ACM, New York, NY, 484–491. DOI: http://dx.doi.org/10.1145/1571941.1572025

Craig Silverman. 2014. *Verification Handbook: A Definitive Guide to Verifying Digital Content for Emergency Coverage*. European Journalism Centre.

David A. Smith and Gregory Crane. 2001. Disambiguating geographic names in a historical digital library. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'01)*, Panos Constantopoulos and Ingeborg Sølvberg (Eds.). Springer-Verlag, London, 127–136.

Russell Swan and James Allan. 1999. Extracting significant time varying features from text. In *Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM'99)*. ACM, New York, NY, 38–45. DOI: http://dx.doi.org/10.1145/319950.319956

Tasnia Tahsin, Davy Weissenbacher, Robert Rivera, Rachel Beard, Mari Firago, Garrick Wallstrom, Matthew Scotch, and Graciela Gonzalez. 2016. A high-precision rule-based extraction system for expanding geospatial metadata in GenBank records. *Journal of the American Medical Information Association* 23, 5 (2016), 934–941. DOI: 10.1093/jamia/ocv172

Dan Tasse, Zichen Liu, Alex Sciuto, and Jason I. Hong. 2017. State of the geotags: Motivations and recent changes. In *Proceedings of the 11th International Conference on Weblogs and Social Media (ICWSM'17)*. 250–259.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The new data in multimedia research. *Communications of the ACM* 59, 2 (2016), 64–73.

Marc Verhagen, Roser Saur, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval'10)*, 57–62.

Xingguang Wang, Yi Zhang, Min Chen, Xing Lin, Hao Yu, and Yu Liu. 2010. An evidence-based approach for toponym disambiguation. In *18th International Conference on Geoinformatics (Geoinformatics'10)*. Article 5567805.

Stefanie Wiegand and Stuart E. Middleton. 2016. Veracity and velocity of social media content during breaking news: Analysis of November 2015 Paris shootings. In *Proceedings of the 3rd Workshop on Social News on the Web (SNOW'16), Companion of the 25th International World Wide Web Conference WWW'16)*.

Benjamin P. Wing and Jason Baldridge. 2011. Simple supervised document geolocation with geodesic grids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT'11)*, Vol. 1. 955–964.

Jiangwei Yu Rafiei and Davood Rafiei. 2016. Geotagging named entities in news and online documents. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM'16)*. ACM, New York, NY, 1321–1330. DOI: https://doi.org/10.1145/2983323.2983795

Wei Zhang and Judith Gelernter. 2014. Geocoding location expressions in Twitter messages: A preference learning method. *Journal of Spatial Information Science* 9 (2014), 37–70.

Received April 2017; revised February 2018; accepted March 2018