Cross-lingual geo-parsing for non-structured data

Judith Gelernter
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
gelern@cs.cmu.edu

Wei Zhang
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
wei.zhang@cs.cmu.edu

ABSTRACT

A geo-parser automatically identifies location words in a text. We have generated a geo-parser specifically to find locations in unstructured Spanish text. Our novel geo-parser architecture combines the results of four parsers: a lexico-semantic Named Location Parser, a rules-based building parser, a rules-based street parser, and a trained Named Entity Parser. Each parser has different strengths: the Named Location Parser is strong in recall, and the Named Entity Parser is strong in precision, and building and street parser finds buildings and streets that the others are not designed to do. To test our Spanish geo-parser performance, we compared the output of Spanish text through our Spanish geo-parser, with that same Spanish text translated into English and run through our English geo-parser. The results were that the Spanish geo-parser identified toponyms with an F1 of .796, and the English geo-parser identified toponyms with an F1 of .861 (and this is despite errors introduced by translation from Spanish to English), compared to an F1 of .114 from a commercial off-the-shelf Spanish geo-parser. Results suggest (1) geo-parsers should be built specifically for unstructured text, as have our Spanish and English geo-parsers, and (2) location entities in Spanish that have been machine translated to English are robust to geo-parsing in English.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: *Linguistic processing*

General Terms

Algorithms, Design

Keywords

geo-parse, location, translation, Spanish, Twitter, microtext, geo-reference, cross-language geographic information retrieval (CL-GIR)

1. INTRODUCTION

Translating a document into a language plentiful in language processing tools and location resources, such as English, amounts to cross-language geo-information retrieval. We will examine Strötgen's view that normalized location information is language-independent [16]. Difficulties in cross-language experiments are exacerbated when the languages use different alphabets, and when the focus is on proper names, as in this case, the names of locations.

Geo-parsing structured versus unstructured text requires different language processing tools. Twitter messages are challenging to geo-parse because of their non-grammaticality. To handle non-grammatical forms, we use a Twitter tokenizer rather than a word tokenizer. We use an English part of speech tagger created for tweets, which was not available to us in Spanish. Our machine learning models for recognizing location words acknowledge Twitter-specific features such as hash tags (index terms specific to Twitter), website addresses and names in Twitter @mentions. Both our English and Spanish models were trained on tweets.

Our focus is social media for crisis response due to the importance of location in message text. The use of social media for disaster relief had been established by the time of the Haiti earthquake in January 2010, when posts to Facebook, Flickr, YouTube and Twitter were numerous [5]. Research revealed that those messages sent during the Haiti earthquake that were considered "actionable" and more useful than the others were generally the ones that included locations [10]. Moreover, Twitterers themselves are more likely to pass along, or re-tweet, messages that include geo-location and situational updates [17], indicating that Twitterers find such messages important. Our data for this research are tweets pertaining to a 2010 earthquake in Chile.

The basis of our experiment is to send a test set of Spanish Chilean earthquake messages through our Spanish geo-parser, a commercial Spanish geo-parser, and in English translation through our own English geo-parser. The output of the three algorithms were scored in comparison to manual geo-tagging annotation of the Spanish tweets by fluent Spanish speakers.

Our English translations of the Spanish tweets come from Google Translate (see Table 1 for example).[1] This tool finds statistically-recurring patterns and selects out of hundreds of millions of human-translated documents what seems to be the best translation for unseen word groupings. Given the vast amount of high quality Spanish-English translations, the Google Translate output is also of high quality.

Our research questions are:

How does our Spanish geo-parser perform on identifying toponyms in Spanish tweets in comparison to our English geo-

GIR'13, November 05 2013, Orlando, FL, USA
ACM 978-1-4503-2241-6/13/11$15.00
http://dx.doi.org/10.1145/2533888.2533943

[1] http://translate.google.com/ and according to Google Translate "About", http://translate.google.com/about/

Table 1. Sample tweet on the February earthquake in Chile, 2010 (with locations in bold)

Spanish tweet	Info geologica Terremoto **Chile**: …construidos sobre la parte plana de **concepcion**, dundadas sobre suelo lando (arenas del **bio bio** y **relleno**.
Google-translated tweet	**Chile** earthquake geological info: … built on the flat of conception, on soil dundadas lando (**bio bio** sands and fillers)

parser finding toponyms on those tweets Google-translated into English?

How does our Spanish geo-parser for unstructured text compare to a commercial geo-parser for Spanish?[2]

Our contributions include (1) a Spanish geo-parser that is tuned to informal, unstructured text,[3] and (2) a prediction based on our experimental results about the relative utility of geo-parsing within a language compared with geo-parsing across languages which use the same alphabet.

2. LITERATURE REVIEW

2.1 Geo-parsing

Geo-parsing is a subset of named entity recognition, which identifies entities in text and classifies them into categories such as agent, organization and location. Our literature review focuses therefore on language processing rather than geographic information retrieval.

Geo-parsing is performed automatically based on language rules. A model based on language rules is then applied to unseen instances to find location words in unseen text [2]. The text of a tweet is not only unstructured, it is also brief. If not supplemented with metadata, it is potentially difficult to parse due to lack of context. It has been found that a data set too small or with insufficient context for the location word as in the geo-parsing of word or phrase search queries is liable to yield low recall and precision results [8].

Our Spanish geo-parser was created specifically for this research. Our English geo-parser was created earlier [6], but it has continued to be developed in concert with our Spanish geo-parser for the benefit of this research. Other geo-parsers include those by MetaCarta, Yahoo! Placemaker, GeoGravy, NetOwl, and open source systems such as CLAVIN, TextGrounder, Geodict, GeoDoc, Geotxt, Europeana, OpenSextant and DIGMAP, as well as the Unlock system from the University of Edinburgh, and an open source geo-parser soon to be available from Leetaru.[4]

2.2 What to geo-parse?

Locations found by geo-parsers are usually toponyms, or political and administrative place names at the resolution of city or higher in the spatial hierarchy. Some have mined for telephone number and postal codes for additional location precision [1], but neither telephone nor postal information appear in tweets with any frequency. We parse for streets and buildings as well as toponyms because that is what our preliminary inquiry into locations in tweet text uncovered [7]. The OpenCalais by Thomson Reuters parses for facility, which we call building.[5]

2.3 Geo-parser structure

Multi-lingual text mining systems have been developed in one language and ported to another, as for example Gamon et al. [4]. For cross-lingual applications such as cross-lingual topic tracking or linking related documents across languages, the most common approach is to use Machine Translation or bilingual dictionaries to translate foreign languages into one language, typically English [15]. We will try this approach for geo-parsing tweets.

2.4 Misspelling

The Levenshtein algorithm that identifies misspellings was proposed in the 1960s, and the original paper is in Russian.[6] The distance between two spellings has been called the "edit distance" in that it is the number of edits needed to transform one string into another string. Edits are insertion, deletion or substitution. Other methods of identifying mis-spelled words have been based on web-wide data with many more words spelled correctly than mis-spelled [18]. This, along with the edit distance of Levenshtein, is the basis of the Norvig algorithm that uses probability theory to find the correct spelling of a word [11.]. Misspelling sub-routines are not standard in geo-parsers, so that locations spelled incorrectly or in non-standard form are often missed.

3. GAZETTEER STRUCTURE

This section describes the structure of our gazetteer as it is part of both the Named Location Parser and the Named Entity Recognizer parser. Gazetteer structure is essential because of its immense multi-GB size. Without accommodating the data structure, gazetteer lookup would make processing time unacceptable.

3.1 Gazetteer as trie and inverted index

We use the structure of a trie tree (from re**trie**val) because it is flexible for information look-up. Searching character by character allows us to match word approximations that are misspellings, and adjectival locations that are similar in stem but differ in ending from the gazetteer term. We can also match with possessive forms and declensions in other languages.[7] The trie can also

---

[2] The identity of the proprietary Spanish geo-parser is not revealed to respect a request from the company.

[3] Our geo-parser has been available on GitHub at https://github.com/geoparser/geolocator

[4] These are found at the following web addresses as of February 7, 2012: Metacarta geoparser at http://www.metacarta.com/products-platform-queryparser.htm, Yahoo Placemaker at http://developer.yahoo.com/geo/placemaker/; the Unlock system http://geogravy.com/ ; http://www.netowl.com/ ; https://github.com/Berico-Technologies/CLAVIN https://github.com/utcompling/textgrounder https://github.com/petewarden/geodict http://geodoc.stottlerhenke.com/geodoc/ http://geotxt.org/ http://europeana-geo.isti.cnr.it/geoparser/geoparsing https://github.com/OpenSextant/opensextant https://code.google.com/p/digmap/wiki/GeoParser at http://unlock.edina.ac.uk/texts/introduction

[5] http://www.opencalais.com/

[6] "Levenshtein distance" entry in Wikipedia, Retrieved February 17, 2013 from http://en.wikipedia.org/wiki/Levenshtein_distance

[7] Pouliquen et al (2004) point out that in Finnish, "London" is spelled "Lontoo" in the nominative, but "Lontoon" in the genative, Lontooseen" in the dative, "Lontoolaisen" for a Londoner, and more. Pouliquen et al, (2004). Geographical information recognition and visualization in texts written in various languages. *Proceedings of the*

recognize some demonyms, which are inhabitants of a particular place (like Chilean for Chile).[8]

Each node contains an array of pointers to every letter of the toponym, and proceeds down the trie. Each toponym can be stored in multiple forms: an English version, a version in the native language, a version in the native language without capitalization or accents, and a version in other major languages. Added to this are the bigram and trigram versions of the name for the misspell parser.

When a word is sent to the gazetteer for lookup, the root of the tree directs it to matching letters. After each letter is matched one by one, it looks to the next letter. Locations might be in multiple languages. The geo-parser finds multiple potential matches at the same time, and outputs as many as match.

3.2 Performance optimization

The geo-parser includes two instances of the gazetteer in two forms: as a trie which is stored in memory, and as an inverted index which is stored on disk. The trie has been incorporated into the Named Entity Recognizer, and this is linked by ID to the inverted index form of the gazetteer. Linking the two helps balance processing speed and memory usage.

For efficiency, we load only a portion of the gazetteer for the Named Entity Recognizer into memory when the algorithm starts. The rest of the gazetteer is indexed by Apache's Lucene,[9] an open-source information retrieval library, and stored on the disk. In order to parse on the fly, we built into memory another instance of the gazetteer as a trie that performs high speed gazetteer lookup. A separate module loads its pre-trained models into the memory, and performs inferences for each tweet based on these models.

4. GEO-PARSING ALGORITHM

The initial system should be modular so that it can adapt easily to other languages and have simple rules, uniform input and output structure, and shared grammars for cases in which features are the same across languages [14]. We agree with Pastra et al. [12] in that "at the most shallow level, if a language or processing resource cannot easily be separated from the framework within which it was developed, it is of little use to anyone not using that system. Also, a rigid system architecture may constrain the way its components can be used, so that even if they are separable, they may require embedding in a similar framework in order to be of use (p. 1412)." The architecture of our Spanish and English geo-parsers follow this principle of modularity.

This section describes separately the mis-spell parser that is part of pre-processing, and then toponym parser, street and building parser that are part of the main algorithm. Data flow through the algorithm is diagrammed in Figure 1.

4.1 Pre-processing

4.1.1 Pre-processing

Tweets. The geo-parsers can accept .json tweet files. The output

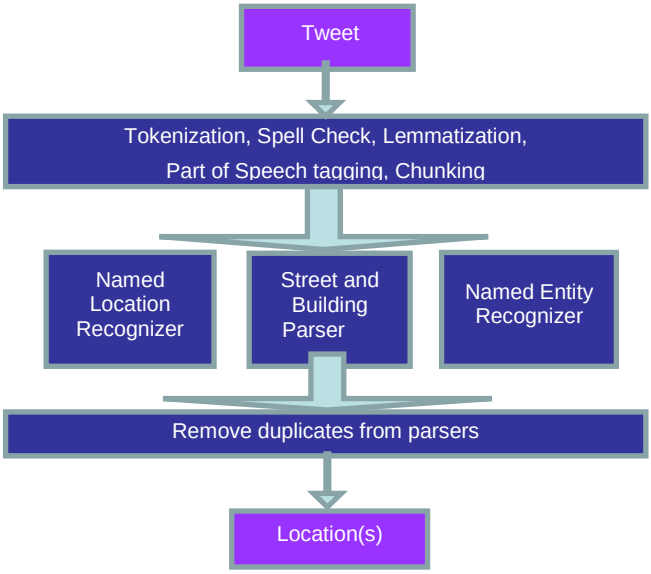

Figure 1 Data flow in the Spanish and English geo-parsers

is tweet + location(s), which is what we have as output from the human annotations. The algorithm first detects the language of the data, and then calls up the correct geo-parser based on the language detected. Language detection is performed by Cybozu, a third party software.[10]

Preprocessing tools. A pre-processing module cascades from tokenization, to spell-correction, to lemmatization,[11] and part of speech tagging. It also loads into memory the gazetteer and the dictionary, building and street lists, and the manually-assembled common words filter for the gazetteer. A dictionary list with locations removed is treated as an additional filter for the gazetteer.

We used a tokenizer specifically for tweets[12] for both Spanish and English. For Spanish, we modified it by adding punctuation of ¿ ¡ as new delimiters. We also remove accents from the data and make all lower case before proceeding. Discovered tokens are sent to spell check (detailed in section 4.1.2) before continuing with the pre-processing, otherwise mis-spelled words will not be lemmatized or part-of-speech tagged correctly.

We used an open source Spanish lemmatizer and part of speech tagger,[13] and the Stanford NLP tools for English.

We use a part of speech tagger developed on English tweets.[14] Adjectives are not handled by the English version of the geo-

2004 ACM SAC, March 14-17, 2004, Nicosia, Cyprus, New York: ACM, 1051-1058.

[8] A limited rule set for making city-inhabitant names from city names in French is discussed in Maurel, D., Piton, O. Eggert, E. (2001). Les relations entre noms propres: Lieux et habitants dans le projet Prolex, in D. Maurel & F. Guenthner: *Traitement automatique des langues 41*(3): Hermes, 2001, 623-641.

[9] http://lucene.apache.org/

[10] http://code.google.com/p/language-detection/

[11] Lemmatization is a more reliable basis for the part of speech tagging than is stemming. The stemmer recovers partial forms, whereas the lemmatizer recovers full forms…for the word "fascinating" – the stemmer recovers partial forms, "fascinate-" and the lemmatizer recovers "fascinate".

[12] The compiled code is on Google Code; http://code.google.com/p/ark-tweet-nlp/downloads/list and the full code is on GitHub https://github.com/brendano/ark-tweet-nlp/

[13] http://code.google.com/p/mate-tools/downloads/list

parser, but they are included in the Spanish version, and they resolve into toponyms. The Spanish lemmatizer includes also part of speech tags.

We built our own chunkers for English and Spanish based on grammar rules. The basic rule is that one or more adjectives plus one or more nouns makes a chunk. Differences in the chunking algorithms come from differences in Spanish and English grammar, and from the different part of speech tags assigned by their respective part-of-speech taggers. We ran the data before and after chunking, and found that the chunking improves results because it reduces false positives.

Gazetteer. We were unable to find a gazetteer in which the place names of the world are exclusively in Spanish. However, countries and major cities in GeoNames are in major languages, including, of course, Spanish. GeoNames is a geographical database with over 10 million names (as of winter 2013).[15] It relies largely on names from the National Geospatial Agency world gazetteer and the U.S. Geological Survey of Geographical Names.

Dictionary to filter the gazetteer. Finding a digital Spanish dictionary available for download proved surprisingly difficult. The dictionaries that are freely available are on-line lookup or are part of other applications (as the spell-checker for a mobile device), or are older dictionaries that have been scanned and uploaded as .pdf (as stored in the Internet Archive[16]). So we created our own list of commonly-found Spanish words based on Spanish web pages from the ClueWeb09 data set.[17] A few Spanish web pages were used as seeds, and other pages linked to those were found using the web crawler Heritrix.[18] Examples of Spanish pages were blog posts and a site listing radio channels. We extracted the content from about 800 Mb of raw data using BoilerPlate.[19] Then we removed numbers, single letters, control characters and punctuation, as well as words in English. The remaining Spanish words found on those pages were then listed in descending order of frequency. From this list, we removed cities and countries. We retained about 80,000 words with which to filter common words from the gazetteer, a list comparable in size to our 100,000-word English dictionary word list.[20]

4.1.2 Spell check

The misspell parser considers nouns that do not match with any words in the dictionary or the gazetteer. The Spanish misspell parser considers adjective forms as well.

Spell correction is not invariably triggered when the system encounters a new out-of-vocabulary word, since that word in a Twitter environment might be an abbreviation. A preliminary solution for this is to examine the length of the out-of-vocabulary word: if the word length is short (say, less than 4 characters), then it is more likely to be an abbreviation. If the length is long, then the misspell procedure is initiated.

The misspell parser uses the trie form of the gazetteer. Similarity is calculated between the word in the data and a word in the gazetteer using the Lucene ranking measure (a form of tf-idf). This gives the top N strings from the Lucene gazetteer word list. For those N strings, we calculate the Levenshtein distance, and find the gazetteer entries with the minimum distance as the possible candidate matches. The algorithm can identify misspelling such as repeated letters (Chilee), transposed letters (Pitstburgh), or omitted letters (Pittsburg).

Table 2 Mis-spelled input and match with the gazetteer

Misspelled Form	Original Form	Mis-spell Type	Score(Ci)
San Jose**s**	San Jose	Extra chararacter at the end	San Jose 5.0
Pen(**n**)sylvan**(i)**a	Pennsylvania	Character missing	pensylvania : 5.0 pennsylvania : 1.5
cali**fr**onia	california	One character is flipped	california : 3.5 fronconia : 0.17 caledonia front :0.1
ca**aaaaa**lifo**ooo**rnia	california	Repetition	Direct match.
pittsburgh	Pittsburgh	Case wrong	Direct match.
bus	Not available in gazetteer	Normal word	Not available in gazetteer
carolin**n**ia	carolina	Combination of errors	carolina : 2.0 caroline : 0.67 colonia carolina : 0.2 the carolinian : 0.17 the carolinian inn : 0.11
Park **santiago**	Santiago park	Word flip. The algorithm does not work.	rio santiago : 0.25 minas santiago : 0.2 santiago park : 0.1 santiago park plaza : 0.06666667

Misspell procedure

If the word in the data is not an ordinary word in the dictionary, then it might be misspelled. The inference function can correct a misspelling in some cases. Here's how it works:

a) Bigram and trigram candidates are generated for gazetteer entries and the dictionary. All the locations in the gazetteer LP={LP_1, LP_2,…,LP_n} and all the dictionary words are indexed with Lucene in the form of n-gram vectors.

b) Features are extracted for ranking the unigram, bi-gram and tri-gram gazetteer entries.

c) Given a target phrase T in data, we find the bi-gram and tri-gram candidates in the gazetteer for similar words and use this as a query to search the index.

d) Lucene's ranking function generates a list of candidates {C_1,C_2,…,C_k} from the gazetteer. The frequency for each C_i is f_i. Frequency is generated by counting the appearance of C_i in the candidate ranking list.

e) A similarity score is generated by calculating the similarity of Target T for each Candidate C_i. The purpose of scoring is to find the most probable candidate as our correct output. We developed a measure called "NTE" (n-gram, tf-idf, and edit distance)

$$\mathbf{NTE_T(C_i) = f_i / L(T,C_i)}$$

where f_i denotes the frequency that a location name appears in the list of candidates from the gazetteer. The higher the frequency, the higher is the score generated and the higher the probability that the candidate is correct. $L(T,C_i)$ denotes the edit distance between Target T and Candidate C_i.

[14] http://github.com/brendano/ark-tweet-nlp/
[15] http://www.geonames.org
[16] http://www.archive.org
[17] http://lemurproject.org/clueweb09/
[18] http://crawler.archive.org/index.html
[19] http://code.google.com/p/boilerpipe/.
[20] Our English list is a free download for Unix systems, see http://en.wikipedia.org/wiki/Words_%28Unix%29

The assumption is that if C_i appears more frequently in the candidate list, and C_i is more similar to T in term of word form, then C_i has a higher probability of being the correct form for T than the other candidates.

f) The highest numerical score in many cases is given to the word in the gazetteer and in the dictionary that is the correct match with the misspelled word. See Table 2 for an example of the score output.

The misspell algorithm could be used for words that are transliterated from other languages, or from words written from voice communication as well. Further experimentation with this sub-procedure is planned.

4.2 Processing

A novelty of our system is that we include four different parsers to identify location words. The Named Location parser uses lexical pattern matching with the trie form of the gazetteer. The Named Entity parser uses machine learning. The rules-based building and street parsers are combined for efficiency, and both parsers rely on street or building indicator words. There is some duplicate output among the Named Location and Named Entity parsers, which we remove before result output. The parsers each have different strengths. The Named Location Parser is strong in precision, and the Named Entity Parser is strong in recall, and the building and street parser finds entities that the others are not set up to do. Here we describe how each parser works.

We used learning models sequentially for the toponym sub-routine of each geo-parser. Dozens of runs of the training data helped us to improve the algorithm. The present state of the Spanish geo-parser and its English cousin are due to what we learned in error analysis of the training data and our subsequent adjustments.

4.2.1 Named Location Parser

This parser uses part of speech and then chunking to identify candidate locations in the data. It searches for full string matches in the trie. If there is no exact match in the trie, but it matches 90% of the string from the beginning, it could be a fuzzy match: an adjective or misspelling. The inverted index form of the gazetteer does not have the complete word to lessen the number of false positives in the output.

Some place names in the gazetteer, such as "Steiner" and "Steiner ranch," have multiple lengths. In these cases, we use the coarse rule of matching the word in the data to the longer gazetteer referent when there are two such options. Here is an example of how this parser matches for "Argentinos" in data:

1. Feed Argentinos to the trie tree → stem to Argentin

2. Find "Argentina" and "Argentinian" in the gazetteer, and both begin argentin. But we use "Argentina" as the matched form, because it is shorter than "Argentinos."

3. If a word in the data has no match in the trie tree, we believe it is not a location or location adjective, or it may be a misspelled common word from the dictionary.

Our filtering process prevents a lot of false matches. However, recall suffers in cases where only part of a place name is found in the data (as when data refers to "Mooncup" for a town named "Mooncup hill"), and where place names are comprised of multiple common words that are filtered out (such as "Blue Farm").

Table 3 Features used for NER training

Actual words (in lemma form) found before and after the location word in a window of size 3 before and after the location word. This includes punctuation, and distance and direction words.
Capitalization of the current word, and capitalization sequence of the left and right window of the current word (size 3) for both left and right
Part of speech – part of speech of the current word, and part of speech sequence of the left and right window of the current word (size 3)
Building or Street – the presence of these indicator words from the street and building list
Preposition – the presence of a preposition one or two words before the current token, with prepositions from our list (that does not include time prepositions such as during or while)
Gazetteer–if the current word appears in the gazetteer

4.2.2 Named Entity Recognition (NER) Parser

We experimented with several NER packages before making a selection. We compared the output of our tweet-trained Conditional Random Fields (CRF) with that of the Voted Perceptron-trained Hidden Markov Model (HMM) and the Stanford CRF-based NER that was trained on a huge quantity of structured data. Informal results of our experiment showed that, in many cases, the three disagree on what is and what is not a location.

We tried numerous models trained on different combinations of features and different parameters for each feature (like context word window size, or part of speech tag window size, or gazetteer tag window size). The best feature combination was selected by a variation of the Latent Dirichlet Allocation (LDA) model called semi-supervised context LDA. Feature selection is an important aspect of the utility of the algorithm. We use the same feature categories of word tokens, part of speech, dictionary and gazetteer features for both geo-parsers (see Table 3)

Initial experiments confirmed [3], that the perceptron Hidden Markov Model (HMM) classified toponyms with higher precision than did the Conditional Random Fields (CRF). But the CRF recall is significantly better than that of the perceptron algorithm. Hence, we used CRF for both the Spanish and English NER parsers.

Our toponym parser for Spanish uses the java implementation of Conditional Random Fields developed at the Indian Institute of Technology, Bombay.[21] Our toponym parser for English uses

[21] http://crf.sourceforge.net/

both the Stanford NER (that is based on Conditional Random Fields) and our Twitter-trained CRF.

4.2.3 Street and Building parsers

The rules for the street and building parsers are based in part on chunking. These rules derive from part of speech sequences such as adjective + noun and use street indicator words and abbreviations such as "calle", "cl", "carreterra", "cr", "cra". We collected these words by mining the training set, and by supplementing the list with the help of a Spanish linguist. The building indicator words were made similarly from training tweets and discussion with a Spanish linguist. In addition, we mined building words from the open-source knowledge base, Freebase, and then translated them to Spanish with the help of our Spanish linguist.[22] The part of speech rules to find candidates for the street and building parser are shown in Table 4. In each rule, the string must satisfy the part of speech type and must include either a street or building indicator. The string must satisfy the part of speech rule, and also the last word must be the building or street indicator word.

5. DATA SET

5.1 Data set statistics

We did not find a suitable pre-collected tweet set, so we collected and annotated our own.[23] We collected by keyword and data a set of crisis-related tweets. Our tweets date between February 27, 2010 (the day of the earthquake), and March 15, 2010, with the time zone Santiago. The keywords used to collect the data are: Terremoto, terremotochile, chillán, Talca, Talcahuano, constitution, Biobío, Maule, Cauquenes, Concepción, Cohquecura, Curanipe, Valparaíso, O'Higgins, La Arqucanía, tsunami. Of the 36,000 tweets that met these date and keyword specifications, about 92% were in Spanish, and the rest were in Portuguese, English and a few other languages. We annotated only those tweets in Spanish. Statistics appear in Table 5.

Table 4 Part of speech regular expressions used to find candidates for street and building parser D=articles; A= adjective; N=noun, Z=number, F=punctuation, V=verb, S= prepositions and postpositions[24]

Street (st) Rules	Building (bg) Rules
\\$+[A(street word)]+[N]	[D]*[A(bg word)]+[N]
[D]*[AN][(st word)]	(bg)S[AN]+[NA]
[D]*[A(st word)][AN][N]	[(bg)+[D][N]
[D]*[A(st word)][AN][AN][N]	(bg)[S][N]
(st word)Z	(bd)SDN
(st word)AFVZ	(bg)SNSN[A]?
(st word)A	[bg][F][S][N]
(st word)DN	[A(bg)]+
[st word][N][Z]* | [N][A][Z]*	(bg)A
[st word][S][AN]{0,2}	

Table 5 Statistics for the data set

Data type	Training set	Test set
Number of tweets	3182	1306
Num tweets that contain location(s)	1945	740
Num. of (repeated) locations	2067	799
Num. of streets	29	8
Num. of buildings	116	38
Num. of toponyms	1695	658

Toponym granularity	
countries	54.17
states/regions	2.18
cities, familiar (capitals or tourist destinations)	4.35
cities, non-familiar	31.42
other: streets and buildings	7.88

5.2 Creating a gold standard

Annotation procedure. We asked people fluent in Spanish to annotate categories of street, building, toponym/natural features. When the data contained cases that could be considered either toponym or building, we annotated them as both. For example:

> @e_sanzana El casino de Los Andes tb <sic> esta cerrado por el terremoto, mejor donen la plata en vez de jugar
>
> Annotation:
> Building [casino de Los Andes]
> Toponym [Los Andes] .

Each tweet was geo-tagged independently by two different people. When the two annotators disagreed, a third annotator selected which location was correct. The gold standard was the adjudicated tweets in which a third person reconciled discrepancies among the two original annotators' tags. The high disagreement score for toponyms shows low annotator reliability, and hence a greater importance on the role of the adjudicator in creating a gold standard data set.

Annotator disagreement. We calculated annotator agreement using the kappa statistic. Kappa takes into consideration that agreement by chance. However, we have not used Cohen's kappa in some of our earlier work [6], because it does not take into consideration those instances in which a tweet is determined by both people not to have a location. In this way, kappa makes the agreement seem much lower than it actually is in an effect known as prevalence [9]. We include the kappa statistic in

[22] http://www.freebase.com/

[23] UTGeo201 is a Twitter geo-data set based on tweets collected from the publicly available Twitter Spritzer feed and global search API around the world from September 4, 2011 through November 29, 2011. To select tweets only with verifiable locations (that come from the GPS of the tweeter's mobile device) there are about 38 million tweets from 449,694 users. However, what is available is a set of tweet IDs only [http://www.cs.utexas.edu/~roller/research/kd/corpus/README.txt]. That means that the set has to be downloaded again, tweet-by-tweet, which is an enormous undertaking giving the per diem limits of data transfer.

[24] Postpositions such as "ahead" and "notwithstanding" in the phrases "pressed ahead" or "these objections notwithstanding"

acknowledgement of the evaluation norm, although it is impoverished by this data circumstance.

To calculate kappa, we used the proportion of locations that agree, *p*, and the locations that would be expected to agree by chance, p_e. The formula is by definition

$$Kappa = \frac{p - p_e}{1 - p_e}$$

Perfect agreement yields a kappa score of 1. We have calculated the kappa scores for each location type separately to examine whether annotators were more likely to agree on some types of locations than others.

Four pairs of people annotated the 4488-tweets, and gives kappa scores between each annotator pair for each location type: street, building, toponym. The average kappa for streets was .623, for buildings .382, and for toponyms .374. Whether because the annotators disagreed on their definitions of location, or because the annotators sometimes skipped locations due to fatigue, the low agreement values even across categories are notable. We should modulate our expectations for toponym identification by the algorithm in light of these statistics showing toponym identification among people.

6. EXPERIMENT and RESULTS

We sent the Spanish test set of tweets through our Spanish geo-parser, and through a commercial off-the-shelf Spanish geo-parser. We scored a location from a geo-parser correct if it matched 60% or more with the gold standard. That is, if two of three words match, we would consider the result a match with the gold standard, as long as the words are in the correct order. But if one of two words matched, we would not consider it correct. Based on this metric, we took recall and precision and F1 measurements.

Our first research question was to determine the viability of our methods for geo-parsing non-structured text in comparison to a commercial off-the-shelf Spanish geo-parser. Our methods fared well, as evidenced by the results (Figure 2). For the same testing set of tweets, our Spanish geo-parser achieved an F1 of .796 on toponyms, in comparison to the F1 of .114 of toponyms for the proprietary Spanish geo-parser.

Our second research question concerned the relative utility of geo-parsing in the native language of the text in comparison to translating the text into English and then using an English geo-parser on the translation. Toponym translation errors introduced into the English version of the Spanish tweets produced 10 faulty toponyms, out of the 799 toponyms in the data. Of these translation errors, 60% were geo/non-geo errors, such as the Chilean city *Concepción* mistranslated as conception (See Table 1). In other cases, even if the tweet was mistranslated so that a place name became a person's name, or a place name was garbled due to a syntax error, there was at least a chance that we could catch these with the toponym parser. But because of the large number of false positives that obtain from this match method, we did not use the entire gazetteer of the world, so place names were missed due to omission. Table 5 on Toponym Granularity shows that 31% of the toponyms were non-familiar cities; these were among the more difficult to parse. Despite errors, our English geo-parser at F1 of .861 out-performed our Spanish geo-parser at F1 of .796 (see Figure 3).

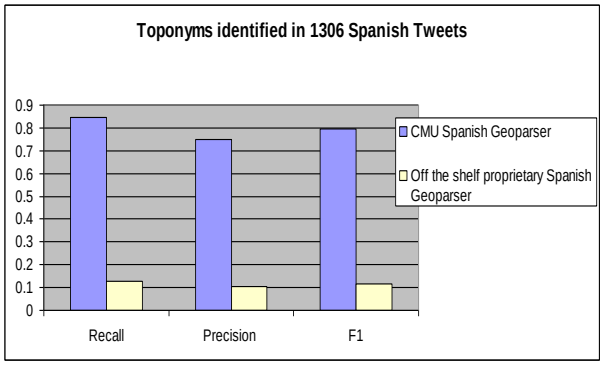

Figure 2. Recall and precision for toponyms for the Spanish tweets in Spanish (for the CMU Spanish and a commercial, off-the-shelf Spanish geo-parser)

What proportion of the accuracy of our Spanish geo-parsing of Spanish toponyms may be attributed to misspelled locations that were corrected? There were 5 misspelled toponyms in the Spanish test set (4 unique toponyms), of which our Spanish geo-parser correctly found 2. Thus, spell correction played a minor role in this data set.

Table 6 Our Spanish Geo-parser on Spanish tweets (N=1306)

Error analysis by toponym granularity		
	recall	**precision**
country	34%	
state/regions	13.63%	
cities, familiar (capitals and tourist locations)	3.13%	
cities, non-familiar	17.33%	78.83%
other (streets and buildings)	31.83%	21.17%

Geo-parsing text in some languages will reflect the completeness of the gazetteer. GeoNames is strong in English exonyms (place names in other languages translated into English). In our test set, the recall for Spanish was 84.58%, and the recall for English was 85.34%. That means that about 85% of the toponyms in Spanish and English could be found in the gazetteer and despite mis-spellings, could be handled by the geo-parser. The recall for English is slightly higher than Spanish because GeoNames's strength is English. Non-familiar cities, streets and buildings were the sources of error, many of which did not appear in the gazetteer and were missed as named entities. The precision is low because of false positives from non-location nouns incorrectly parsed as locations.

There was apparently insufficient data with sentence context to train the NER Spanish and English parsers for streets and buildings, which hurt recall. Our rules were ineffective because building or street indicator words were present rarely. The NER portion of our Spanish geo-parser performed moderately on building recognition, with .733 precision but only .029 recall, for a combined F1 for buildings of .325. Also the NER portion of our English geo-parser on the Google-translated Spanish tweets for

buildings performed poorly, with a precision of .395, a recall of .366, and a combined F1 for buildings of .380.

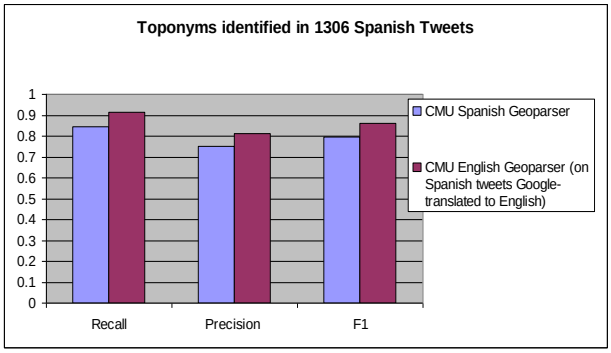

Figure 3. Precision, recall and F1 for toponyms for the Spanish tweets sent through the CMU Spanish geoparser, and those same Spanish tweets translated to English and sent through the CMU English geoparser.

7. CONCLUSION

Our Spanish geo-parser on Spanish tweets performed comparably to our English geo-parser on English translations of the same Spanish tweets. This may be explained in part by the quality of the English geo-parser, which includes resources better adapted to non-structured text, and which includes an additional Named Entity Recognition algorithm to identify location words. We conclude that Spanish named entities such as locations survive machine translation, and may be geo-parsed successfully in quality translation, given effective geo-parsing tools such as our geo-parser for English. This was found also for a test on named entities generally in Arabic and Swahili [13]. The utility of machine translation is especially important for languages with fewer language processing tools for which building a geo-parser would be quite time consuming.

Unlike the majority of geo-parsers, our Spanish and English geo-parsers were built to handle non-structured text, meaning that they can manage words that are out of vocabulary or mis-spelled, as well as non-grammatical forms lacking the proper capitalization or punctuation. This paper describes a method for building such a geo-parser, with an architecture that permits efficient access to the huge knowledge resource of the gazetteer, and a method to handle discrepancies between the spelling of toponyms in data and the standard spellings in a gazetteer.

8. ACKNOWLEDGEMENTS

This research was supported by a grant from the U.S. Army Research Office as part of the DARPA Social Media in Strategic Communications program. We have had on-going talks with Nick Pioch of Systems and Technology Research under the grant, and an earlier draft of the paper benefited from his comments.

9. REFERENCES

[1] Angel, A., Lontou, C. and Pfoser, D. (2008). Qualitative geocoding of persistent web pages. *ACM GIS '08, November 5-7, 2008, Irvine, CA, USA*, [10 p.] Retrieved February 11, 2013 from http://queens.db.toronto.edu/~albert/docs/aelp-gis08.pdf

[2] Batista, D. S., Silva, M. J., Cuoto, F. M., Behera, B. (2010). Geographic signatures for semantic retrieval. *GIR'10 18-19 February 2010, Zurich, Switzerland*, [8 p.]

[3] Carlson, A., Gaffney, S. and Vasile, F. (2009). Learning a Named Entity Tagger from Gazetteers with the Partial Perceptron. Appeared at the *2009 AAAI Spring Symposium on Learning by Reading and Learning to Read*. Retrieved February 10, 2013 from http://www.cs.cmu.edu/afs/cs.cmu.edu/Web/People/acarlson/papers/carlson-aaaisymp-lbr09.pdf

[4] Gamon, M., Lozano, C., Pinkham, J., & Reutter, T. (1997). Practical experience with grammar sharing in multilingual NLP. In *Proceedings of ACL/EACL,Madrid, Spain*, 49–56.

[5] Gao, H., and Barbier, G. (2010). Harnessing the Crowdsourcing Power of Social Media for Disaster Relief. *IEEE Intelligent Systems 26*(3), 10-14

[6] Gelernter, J. and Balaji, S. (2013). An algorithm for local geoparsing of microtext. *GeoInformatica, 17*(4),635-667.

[7] Gelernter, J. and Mushegian, N. (2011). Geo-parsing messages from microtext. *Transactions in GIS 15*(6), 753-773.

[8] Guillén, R. (2007) GeoParsing web queries. In C. Peters et al. (Eds): *CLEF 2007, LNCS 5152*, pp. 781-785.

[9] Gwet, K. (2002). Kappa statistic is not satisfactory for assessing the extent of agreement between raters, http://www.agreestat.com/research_papers/pappa_statistic_is_not_satisfactory.pdf

[10] Munro, R. (2011). Subword and spatiotemporal models for identifying actionable information in Haitian Kreyol. *Fifteenth Conference on Computational Natural Language Learning, CoNLL-2011, June 23-24, 2011*.

[11] Norvig, Peter. (n.d.) How to write a spelling corrector. Retrieved February 17, 2013 from http://norvig.com/spell-correct.html

[12] Pastra, K., Maynard, D., Hamza, O., Cunningham, H., & Wilks, Y. (2002). How feasible is the reuse of grammars for Named Entity Recognition? In *Proceedings of LREC (pp. 412–1418). Las Palmas, Spain*.

[13] Shah, R., Lin, B., Gershman, A., Frederking, R. (2011). SYNERGY: A Named Entity Recognition System for Resource-scarce Languages such as Swahili using Online Machine Translation. *Proceedings of LREC Workshop on African Language Technology*.http://www.cs.cmu.edu/~encore/synergy.pdf

[14] Steinberger, R. (2012). A survey of methods to ease the development of highly multilingual text mining applications. *Language Resources and Evaluation 46*(2), 155-176

[15] Steinberger, R., Ombuya, S., Kabadjov, M., Pouliquen, B., Della Rocca, L., Belyaeva, J., de Paola, M., Ignat, C., van der Goot, E. (2011). Expanding a multilingual media monitoring and information extraction tool to a new language: Swahili. *Language Resources & Evaluation 45*, 311-330.

[16] Strötgen, J., Gertz, M., Junghans, C. (2011). An event-centric model for multilingual document similarity. *SIGIR'11, July 24-28, 2011, Beijing, China*, 953-962.

[17] Vieweg, S., Hughes, A.L., Starbird, K., Palen, L. (2010). Microblogging during two natural hazards events: What Twitter may contribute to situational awareness. *CHI 2010, April 10-15, Atlanta, CA, USA*, 1079-1088.

[18] Whitelaw, C., Hutchinson, B., Chung, G.Y., Ellis, G. (2009). Using the web for language independent spellchecking and autocorrection. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Singapore, August 6-7, 2009*, 890-899.