MapIt: a case study for location driven knowledge discovery and mining

Satyen Abrol and Latifur Khan*

Department of Computer Science,
Erik Jonsson School of Engineering and Computer Science,
University of Texas at Dallas,
P.O. Box 830688, EC 31, 75083-0688, Richardson, TX, USA
Fax: (972) 883-2349
E-mail: abrol@utdallas.edu
E-mail: lkhan@utdallas.edu
* Corresponding Author

Fahad T. Bin Muhaya

Management Information System Department,
College of Business Administration,
King Saud University,
P.O Box 2459 Riyadh, Saudi Arabia
E-mail: fmuhaya@ksu.edu.sa

Abstract: In the present world scenario, everybody is on the lookout for suitable housing options, each having different needs (e.g., the elderly are looking for safe, quiet neighbourhood, while students are looking for affordable apartments close to the university/school). For e.g., Craigslist currently does not have a map version, making the process of apartment searching a very long and laborious process. This creates a need for software that is significantly superior to current web search tools. We demonstrate the development of a tool which takes the Craigslist apartment listings on Google Maps. MapIt then integrates this functionality with the information collected from location based extraction of various web sources such as the city police blotter which makes apartment searching simpler and faster, helping the user to make a better decision. The paper also discusses the challenges that are faced in the development process, the raw and unstructured nature of the documents, the existence of geo/non-geo and geo-geo disambiguities and our approach in identifying the location of the apartment from informal text (geo-parsing and geo-tagging of content) to ensure maximum coverage of the listings.

Keywords: information retrieval; text mining; natural language processing; gazetteer; geo-parsing; disambiguation; geo-tagging.

Reference to this paper should be made as follows: Abrol, S., Khan, L. and Bin Muhaya, F.T. (2013) 'MapIt: a case study for location driven knowledge discovery and mining', *Int. J. Data Mining, Modelling and Management*, Vol. 5, No. 1, pp.57–75.

Biographical notes: Satyen Abrol is a PhD candidate at the University of Texas at Dallas working in the data mining group. His broad research interests are location mining and location based data mining in large scale online social networks where he seeks to understand the relation between location of a user and his social behaviour. He received his Masters in Computer Science from the University of Texas at Dallas and previously Bachelors in Computer Science from Panjab University, Chandigarh, India in 2008.

Latifur Khan received his PhD in Computer Science from the University of Southern California (USC) in 2000. He is currently an Associate Professor in the Computer Science Department at the University of Texas at Dallas (UTD). His research areas cover data mining, multimedia information management, semantic web, and database systems with the primary focus on first three research disciplines. He has served as a committee member in numerous prestigious conferences, symposiums, and workshops, including the ACM SIGKDD Conference on Knowledge Discovery and Data Mining. He has published more than 160 papers in journals and conferences. He is a senior member of the IEEE and ACM.

Fahad T. Bin Muhaya is an Associate Professor and an Information Security Consultant. Currently, he is the Chairman of MIS Department at King Saud University and the Director of His Royal Highness Prince Muqrin Chair for IT Security (PMC). In addition, he is a co-founder of the Center of Excellence in Information Assurance (CoEIA) in King Saud University and was the Vice Director. Also, he has extremely participated and organised several conferences, workshops and has published several journal and conference papers.

1 Introduction

The task of identifying the correct location of documents such as e-mails, news, web pages etc. has always been greatly beneficial for the purposes of data mining and information retrieval. Shekhar et al. (2003) in their book, talk about the diverse list of professionals that benefit from such services. These include mobile phone users looking for nearest gas stations or farmers looking for ways to minimise the use of pesticides on his farm.

In the current era which is witnessing exponential growth of social media, location extraction methodologies can be broadly categorised into two approaches. The first approach focuses on identifying the location from the content of the text in the web page or the user messages (Hecht et al., 2011). The second approach establishes the relationship between geospatial proximity and friendship and predicts the location on the basis of the social network of the user (Abrol et al., 2010).

Geo-parsing is the process of determining geographic coordinates of textual words and phrases that occur in unstructured content, such as 'six miles east of Paris'. You can also geo-parse location references from other forms of media, e.g., audio content in which a speaker mentions a place. With geographic coordinates the features can be mapped and entered into geographic information systems. Once the coordinates are identified the applications plot the geo-parsed text on to a map.

Geo-parsing goes beyond geo-tagging (or geo-coding) as it deals with ambiguous and unstructured text. There are two types of ambiguities that exist: Geo/non-geo and

Geo/geo ambiguities. Geo/non-geo ambiguity is the case of a place name having another, non-geographic meaning, e.g., Paris might be the capital of France or might refer the socialite, actress Paris Hilton. Geo-geo ambiguity arises from the two having the same name but different geographic locations, e.g., Paris is the capital of France and is also a city in Texas. Smith et al. (2001) report that 92% of all names occurring in their corpus are ambiguous.

Researchers have used a variety of methods to tackle the problem of correctly geo-parsing the documents at a city level. In the domain of NLP, the techniques of machine learning are employed to identify the location from their structure and context. We take the Craigslist advertisements consisting of raw unstructured text, and identify locations from them. However, the extracted locations are ambiguous and we use our heuristic-based algorithm to identify one single correct location.

In our work (Abrol et al., 2009), we have made several contributions. First, MapIt is a tool that facilitates the display of Craigslist advertisements onto Google Maps along with other relevant information. This integrated information is obtained from aggregation and analysis of data from POI databases and police blotters in an efficient and timely manner. Second, we devise an efficient algorithm and heuristics to identify and disambiguate the correct location from the unstructured text of the Craigslist advertisement. The algorithm goes beyond the previous work to identify the location up to the street level with 85% accuracy. Finally, we have developed a fully functional prototype and tested on real dataset collected from the Craigslist website.

We also compare our results to those obtained from using a gazetteer-based approach such as Amitay et al (2004). The results show a remarkable improvement in the precision and recall by using the heuristic-pruning-based algorithm. It is significant to note the importance of a higher precision and recall value. A higher precision value guarantees the correctness of the algorithm in identifying the location and a higher recall value makes sure that no ad with a location is left behind.

The research paper is organised as follows. Section 2 surveys and compares the related work in this domain. Section 3 analyses problems with *Craigslist* ads as a source of geo-information. Section 4 describes the types of ads and the technical challenges faced involved in each of them. Section 5 discusses the disambiguation algorithm, and describes how it identifies and disambiguates the location in the text. Section 6 and 7 show how we associate a confidence value with the location and the integration of web sources respectively. Section 8 and 9 discusses the results and concludes with some pointers for the future work.

2 Related works

Geo-geo disambiguation is a topic on which a lot of research has been done. But almost all of the research is restricted to the city, state level disambiguation. Whereas, in our case, the granularity goes one step ahead and we identify up to the street level. First we discuss these city level disambiguation algorithms and then we compare MapIt with some websites that do similar Craigslist Google Maps mash-up, and how our product out performs them.

Like we mentioned before, the identification of location of a webpage can be done using two methodologies. The first one focuses on mining the location from the social network of the webpage or an online social networks user (Abrol et al., 2010). In other

words, the approach hypothesises that there is two web pages are likely to refer to the same location if there exists a link between them (one web page has a link pointing to the other). The other approach is the classical approach of extracting location from the content of the webpage or the content of the messages posted by the user (Smith et al., 2001). In *Craigslist* advertisements, the social graph is nonexistent since no ad usually refers to other ads, in our current work we rely on the content-based geo-tagging approach. Another problem in using the social graph-based approach is that good for city level identification at best. Whereas, our work aims to locate apartments at a much finer granularity, that of street level. Therefore, in our experiments, we compare our results only to the existing content-based geo-tagging approaches described below.

The problem of geographic location identification and disambiguation from the content of the text has been dealt with mostly two approaches. One is involving the concepts of machine learning and NLP and the other is using data mining approach with the help of gazetteers.

In NLP and machine learning a lot of previous work is done on the more general topic of named entity recognition (NER). Most of the work makes use of structured and well-edited text from news articles or sample data from the conferences.

Most research work relies on NLP algorithms and less on machine learning techniques. The reason for this is that machine learning algorithms require training data that is not easy to obtain. Also, their complexity makes them less efficient as compared to the algorithms using the gazetteers.

Other researchers use a 5-step algorithm, where the first two steps of the algorithm are reversed. First, only terms appearing in the gazetteer are short listed. Next, they use NLP techniques to remove the non-geo terms. Li et al. (2002) report a 93.8% precision on news and travel guide data.

McCurley (2001) analyses the various aspects of a web page that could have a geographic association, from its URL, the language in the text, phone numbers, zip codes etc. Names appearing the text may be looked up in White Pages to determine the location of the person. His approach is heavily dependent on information like zip codes etc. and is hence successful in the USA, where it is available free but is hard to obtain for other countries. Their techniques rely on heuristics and do not consider the relationship between geo-locations appearing in text.

The gazetteer-based approach relies on the completeness of the source and hence cannot identify terms that are not present in the gazetteer. But on the other hand they are less complex than NLP, and machine learning techniques are hence faster.

Amitay et al. (2004) present a way of determining the page focus of web pages using the gazetteer approach and after using techniques to prune the data. They are able to correctly tag individual name place occurrences 80% of the time and are able to recognise the correct focus of the pages 91% of the time. But they have a low accuracy for the geo/non-geo disambiguation.

Lieberman et al. (2007) describe the construction of a spatio-textual search engine using the gazetteer and NLP tools, a system for extracting, querying and visualising textual references to geographic locations in unstructured text documents. They use an elaborate technique for removing the stop words, using a hybrid model of *Part-of-Speech (POS)* and *Named-Entity Recognition* tagger. POS helps to identify the nouns and NER tagger annotates them as person, organisation, and location. They consider the proper nouns tagged as locations. But this system doesn't work well for text where name of a person is ambiguous with a location, e.g., Jordan might mean Michael Jordan, the

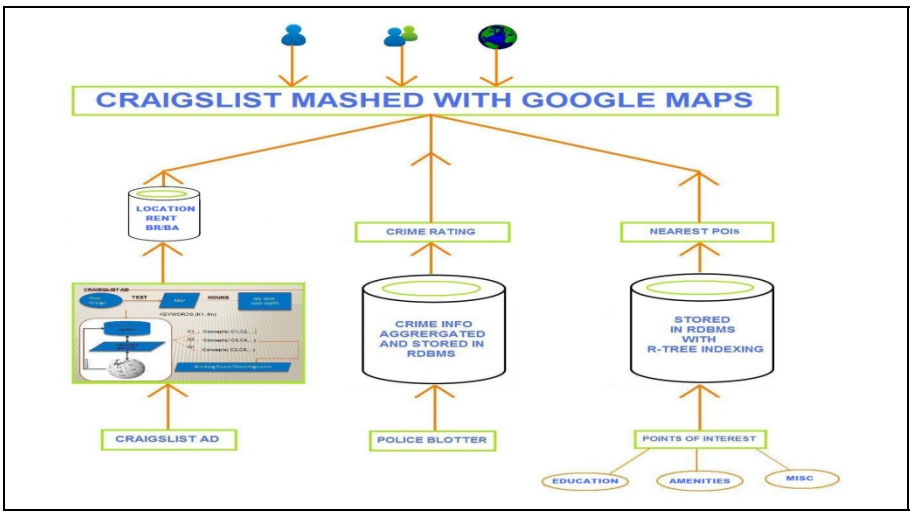
basketball player or it might mean the location. For removing geo-geo ambiguity they use the *pair strength* algorithm. Pairs of feature records are compared to determine whether or not they give evidence to each other, based on the familiarity of each location, frequency of each location, as well as their document and geodesic distances. They do not report any results for accuracy of the algorithm so comparison and review is not possible.

Our approach, in addition to using the *Named-Entity Recognition* tagger, uses a heuristic-based technique that comes up with scores for individual concepts. Each heuristic is designed to ensure that the geo-concepts-based concepts get a higher score boost as compared to the non-geo concepts like names of people, companies, etc.

It is very important to note here that all these strategies and their results are for city level identification and disambiguation. As we later show algorithm makes use of a similar approach, but in addition to it we apply some heuristic techniques that focus only on street level identification. Later we show the impact the addition of heuristics has on the results.

Our previous work (Abrol et al., 2009) was published in the Workshop on Querying and Mining Uncertain Spatio-Temporal Data and discusses the work in progress for extraction of location from *Craigslist* ads. We have extended the work in the workshop publication to include an actual *Craigslist* ad and how the proposed algorithm is able to correctly identify the street concept in the ad.

Figure 1 Architecture of the Craigslist-Google Maps system (see online version for colours)



Craigslist acts as a medium for realtors and owners/renters for free and easy interaction. Existing apartment lookup sites are not dynamic and will not show an apartment or house that has been vacated very recently and is for rent/sale. Nor will it show any special deals that the realtor is offering. Previous attempts to mash Craigslist and Google Maps (Padmapper, Housingmaps, MapKreig), focus only on the graphical interface and functionality, and lack a sophisticated location extraction and hence are not able to display it or fail to do so accurately. These sites only display the listings having the Google/Yahoo maps link in it and use no geo-tagging technique. Other sites such as http://www.allurstuff.com (Allurstuff) do a good job of giving a measure of accuracy but pick only the address that comes after the term 'Location' in the advertisement. In

addition to these there is no site that integrates data sources to provide useful other meaningful information like crime, points of interest etc.

3 Cragslist and its problems

Craigslist is a centralised network of online communities, featuring free online classified advertisements – with sections devoted to jobs, housing, personals, for-sale, services, community, gigs, résumés, and discussion forums (Hover).

Currently Craigslist does not support a map version. So, if someone is looking for an apartment near a particular location, he/she would have to browse through hundreds of listings manually before one can come across a good potential apartment. This makes the process of apartment searching a long and unpleasant task. In addition to this, the user has to separately look on the internet for the other things that affects his decision like crime information, median family income, nearest grocery stores, religious places, hospitals etc. This creates a need for a tool that displays the Craigslist ads on the Google Maps, integrated along with crime statistics, school information, and other points of interest (POIs), so that it becomes easier for the user to make a decision.

Craigslist ads consist of text that is unstructured and have a lot of grammatical and spelling errors. Therefore, it becomes more difficult to identify and disambiguate the location of the apartment/house.

Figure 1 illustrates the architecture of our system. Left most entry of the architecture shows the processing and storing of essential information from an advertisement. The middle and right databases store the crime information and the location based Points of Interests (POIs).

4 Technical challenges

As mentioned earlier, the Craigslist ads consist of unstructured data, usually having a location embedded in the text. Here, we describe six scenarios into which all of the Craigslist ads can be broadly categorised. We then describe how we deal with each of them so as to identify and disambiguate the location.

4.1 Ads with physical address (APA)

APA contains ads which have a complete physical address mentioned with house number, street name and zip. The ads with Google/Yahoo map links also fall in this category. For such ads the location extraction is done through regular expression matching and these ads usually have a CAF value (see Section 5).

4.2 Ads with just one street name (ASN)

ASN consists of the ads that have a street name or a location embedded in the usual unstructured text. It is for this case that we use the disambiguation algorithm, to identify the potential location of the apartment. The ads in this category in the absence of a block number fall in the medium or low confidence level category.

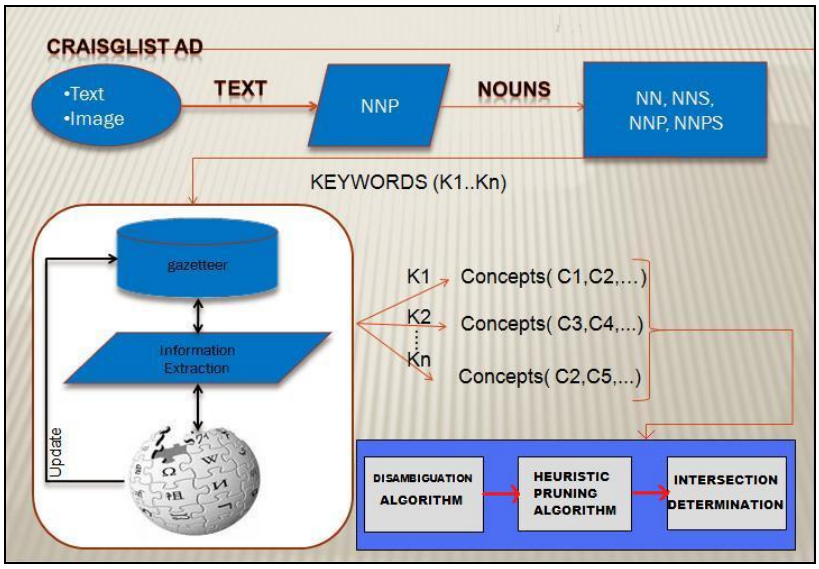
4.3 Ads with intersections (AwI)

Sometimes, the ad publisher describes the location of the apartment as 'near A and B' or 'A at B', where 'A' and 'B' are the street names. If the disambiguation algorithm returns two different street names with comparable weights and close proximity to each other, we check for the streets in the intersection database, for the possibility of an intersection and its coordinates.

4.4 Ads with just phone numbers (APN)

Ads with just a phone number and no mention of the street name or intersection are located using the White Pages reverse lookup. We get the location of the person to whom the phone is registered. Such ads associated with very low values of confidence since we have no proof whether the address is of the realtor or the actual apartment location. The same strategy is used to boost up the confidence level for ASN and AwI ads (see Disambiguation Algorithm for details).

Figure 2 Diagram showing the process of identification and disambiguation of locations (see online version for colours)



4.5 Ads having just neighbourhood (AwN)

There is a major portion of the ads that has just the name of the neighbourhood such as Uptown, Downtown, and Turtle Creek etc. This can help us in narrowing down the area and we can increase the accuracy of the location. We search only the ads where the algorithm returns no address. We maintain a table of all popular neighbourhoods for each city, created from information extracted from *Wikipedia* listings. We search the ads for the neighbourhoods obtained from this table and on a match allocate the location of the apartment as the neighbourhood. This also means a low CAF value (see Section 5) as

compared to a physical address. This is especially helpful to users who are looking for an apartment in a particular area or neighbourhood.

4.6 Ads with no information (ANI)

This section is formed by ads where there is no mention of any street name or potential address, does not have a phone number or it is a mobile or unpublished number. For such ads, the identification of an accurate location is not possible and we just specify the city as the location.

5 The data sources

Before we discuss the heuristic-based disambiguation algorithm in detail it is very essential to describe the various data sources that the algorithm uses.

5.1 TIGER dataset – an open source gazetteer

TIGER which stands for Topologically Integrated Geographic Encoding and Referencing system is an open source gazetteer consisting of topological records and shape files with coordinates for counties, zip codes, street segments, etc. for the entire USA. TIGER/Line Shapefiles are designed for use with geographic information system (GIS) software. The TIGER/Line Shapefiles do not include demographic data, but they contain geographic entity codes that can be linked to the Census Bureau's demographic data.

5.2 Wikipedia

Wikipedia is a multilingual, web-based, free-content encyclopedia project based mostly on anonymous contributions. For our algorithm we used Wikipedia as a source for list of all cities and neighbourhoods that are present in a particular metropolis, e.g., the Dallas-Fort Worth metroplex consists of a large number of cities like Plano, Richardson etc. and some popular neighbourhoods like Turtle Creek, Legacy etc. and to resolve the ambiguities between neighbourhoods and street names, this information is very essential.

5.3 Traffic count data

The traffic count data is the collection of volume of vehicles that pass through some popular streets of a particular city. We make an assumption here that the traffic count of a street is directly proportional to its popularity.

5.4 Important landmarks database (ILD)

The important landmarks database (ILD) is a dataset comprising of all important points of interests like shopping centres, lakes, colleges and universities that are used by the algorithm to identify the location of the apartment.

5.5 Commonly occurring words (COW)

The commonly occurring words (COW) is a collection of all the words that occur very frequently in apartment-based ads with their frequency count, e.g., words like lease, kitchen etc. are very frequent and in most cases will not point out to a location.

6 The heuristic-based algorithm

The algorithm LocationFinder (Ads) is divided into several steps. In this section we describe each of the steps that go into the process of identification and disambiguation of the apartment location.

Algorithm 1 LocationFinder (Ads)

```
Input: Set of ads for determining location
Output: location of the each of the ads
1    for each ad A ε Ads
2        City ← City A
3        Neighbourhood ← Neighbourhood (A)
4        (S, W) ← Street_Disambiguation (A)
5        For each S_i ε S do
6            W_i-new ← Pruning_H (S_i)
7    (S_1, S_2, R) ← Intersect (S, W) /* Pick two intersecting streets with highest weights with
     close mutual weights */
8        If (R==true) then
9            location ← LatLong(S_1, S_2)
10           else location ← S_max-weight (S, W)
```

In line 2 and 3, for each ad, we determine the cities and the neighbourhoods using fixed string matching. From Wikipedia we have the complete set of cities and neighbourhoods for that metropolis. If a term occurs in the cities (or the neighbourhood) list and is not followed by a street suffix such as St., Ave., Rd. etc. then we tag the term as a city (or neighbourhood). Next, we call the method, Street_Disambiguation that helps to identify and partly disambiguate the locations. The method returns the vector containing all possible street names with their weights. In step 5, we call the pruning and heuristic algorithm, Pruning_H, to boost the score. After all the boosting has been done, we first check for the possibility of an intersection. For this we pass the whole vector to a method Intersect which returns true with the street names, S_1 and S_2; false otherwise. In case of an absence of an intersection, we choose the street with the maximum weight to be the location.

We now describe the Street_Disambiguation method in detail. The first step of the method involves removal of all those words from the Craigslist text that are not references to geographic locations. For this, we use the CRF tagger, which is an open source tagger for English with an accuracy of close to 97% and a tagging speed of 500 sentences per second (CRF Tagger). The CRF tagger identifies all the proper nouns from

the text and term them as keywords $\{K_1, K_2,\ldots,K_n\}$. In the next step, the TIGER dataset is searched for identifying the street and city names from amongst them (TIGER gazetteer).

Algorithm 2 Street_Disambiguation (A)

```
Input: A: Craigslist ad
Output: Vector (S, W): streets and weights vector
for each keyword, K_i      //Phase 1
    for each S_j ∈ K_i   //S_j-Street Concept
        for each T_f ∈ S_j
            type    Type (T_f)
            If (T_f occurs in A) then W_Sj      W_Sj + W_type
for each K_i     //Phase 2
 for each S_j ∈ K_i
    for T_f ∈ S_j , T_s ∈ S_L
        If (T_f = T_s) and (S_j ≠ S_L) then
            type    Type (T_f)
                Weight_Sj    Weight_Sj + W_type
return (S, W)
```

We search the TIGER gazetteer for the concepts $\{C_1, C_2,...C_n\}$ pertaining to each keyword. Now our goal for each keyword would be to pick out the right concept amongst the list, in other words disambiguate the location. For this, we use a weight-based disambiguation method. In Phase 1, we assign the weight to each concept based on the occurrence of its terms in the text. Specific concepts are assigned a greater weight as compared to the more general ones. In Phase 2, we check for correlation between concepts, in which one concept subsumes the other. In that case the more specific concept gets the boosting from the more general concept. If a more specific concept C_i is part of another C_j then the weight of C_j is added to that of C_i.

6.1 Heuristic-based pruning and boosting

The heuristic-based algorithm consists of two phases. The first phase is the pruning phase in which we remove all those concepts which are already identified as city or neighbourhood. Next, on the remaining concepts we apply the nine heuristics that either boost up or decrease the weight of the concept.

6.1.1 Commonly occurring words (COW)

As mentioned before we have a created database of commonly occurring words for the apartment/housing domain. If a concept is found in the database, this means the word is commonly occurring and in most likelihood does not correspond to a location (like kitchen, swimming pool etc.) And we deduct the score by a number which is proportional to the frequency of the word.

6.1.2 City distance

The next step is to check the distance between the concept and the city in the original ad. If the two occur very close to each other, the score is boosted.

Algorithm 3 Pruning_H (Si)

Input: Concept Si

Output: Boosted weight Wi of the concept Si

1 If (S_i==City) then return $W_i \leftarrow 0$

2 If (S_i==Neighbourhood) then return $W_i \leftarrow 0$

3 If (N_Gram(S_i)==Semantic(located)) then $W_i \leftarrow W_i+W_1$

4 If (Prev_Word(S_i)==Digit_Sequence) then $W_i \leftarrow W_i+W_2$

5 If (After_Word(S_i)==Street_suffix) then $W_i \leftarrow W_i+W_3$

6 If (S_i ε COW_Dataset) then $W_i \leftarrow W_i-W_{COW\text{-}freq}$

7 If (Dist (S_i,City)<d_0 then W $\leftarrow W_i+W_3*(d_0-$ Dist((S_i,City))

8 If (S_iεTraffic) then $W_i \leftarrow W_i+W_4*$Traffic_Count(Si)

9 If (S_i==Reverse_Phone_Street) then $W_i \leftarrow W_i + W_{RS}$

10 If (S_i=Name_Phone_Street) then $W_i \leftarrow W_i + W_{NL}$

11 If (Distance (S_I, POI) < d_0) then $W_i \leftarrow W_i+W_{POI}$

12 return (S_i,W_i)

6.1.3 Traffic count

In the real world scenario, a home owner would advertise his apartment saying it is located 'some blocks from ABC' where ABC is either some important landmark or some popular street. To utilise this, we collected data of all the popular streets in the cities and their traffic count. We assume here that the traffic count is directly measure of the popularity of the street. A higher traffic count means a higher weight boosting.

6.1.4 Name lookup

Then, we look for any names of people mentioned in the ads, look it up on White Pages and check, if the street mentioned is same as the concept then the score is boosted accordingly.

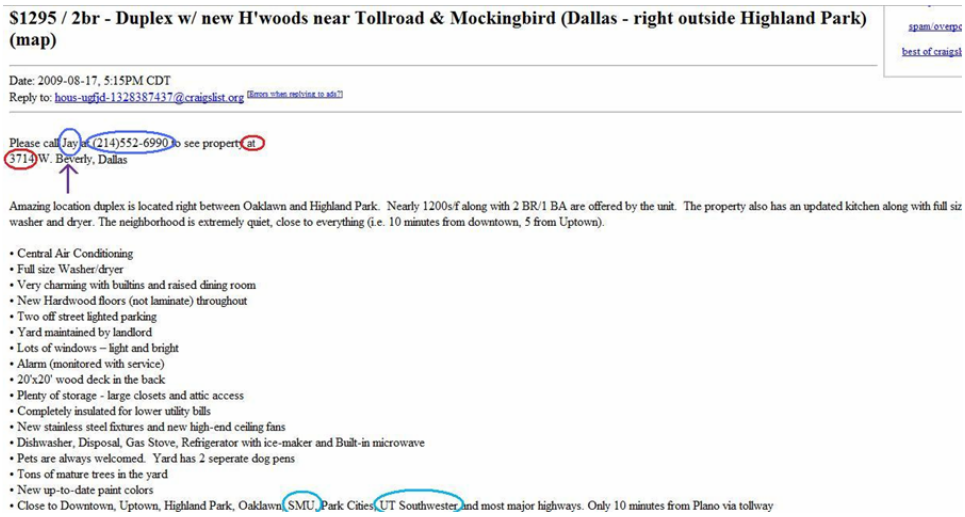
6.1.5 Reverse phone lookup

In the next step we search for phone numbers mentioned in the ads; do a reverse lookup on White Pages and check, if the street mentioned is same as the concept then the score is boosted accordingly.

6.1.6 N-gram analysis

We apply N-gram analysis and consider five words before and after the concept in the original ad. If any of the word is semantically similar to the 'location' the score is boosted.

Figure 3 Example of a *Craigslist* ad where suitable keywords are highlighted to show the application of the heuristics (see online version for colours)



$1295 / 2br - Duplex w/ new H'woods near Tollroad & Mockingbird (Dallas - right outside Highland Park) (map)

Date: 2009-08-17, 5:15PM CDT
Reply to: hous-ugfjd-1328387437@craigslist.org

Please call Jay at (214)552-6990 to see property at
3714 W. Beverly, Dallas

Amazing location duplex is located right between Oaklawn and Highland Park. Nearly 1200s/f along with 2 BR/1 BA are offered by the unit. The property also has an updated kitchen along with full siz washer and dryer. The neighborhood is extremely quiet, close to everything (i.e. 10 minutes from downtown, 5 from Uptown).

- Central Air Conditioning
- Full size Washer/dryer
- Very charming with builtins and raised dining room
- New Hardwood floors (not laminate) throughout
- Two off street lighted parking
- Yard maintained by landlord
- Lots of windows – light and bright
- Alarm (monitored with service)
- 20'x20' wood deck in the back
- Plenty of storage - large closets and attic access
- Completely insulated for lower utility bills
- New stainless steel fixtures and new high-end ceiling fans
- Dishwasher, Disposal, Gas Stove, Refrigerator with ice-maker and Built-in microwave
- Pets are always welcomed. Yard has 2 seperate dog pens
- Tons of mature trees in the yard
- New up-to-date paint colors
- Close to Downtown, Uptown, Highland Park, Oaklawn, SMU, Park Cities, UT Southwester, and most major highways. Only 10 minutes from Plano via tollway

6.1.7 Digit sequence

Next we check the word immediately preceding the concept. If the word is a digit sequence then we boost up the score.

6.1.8 Street suffix

Similarly we check the word after the concept in the original advertisement. If the word is street suffix like St., Ave. etc. then a strong boosting to the weight is made.

6.1.9 Distance from POI

Finally we search the ad for POIs using our POI database, obtain the Lat/Long and check the distance of the concept from this POI. The close proximity of the landmarks (like shopping malls, lakes etc.) to the concept means a higher probability and hence a better weight boost.

6.2 Heuristic-based pruning – an example

Figure 3 shows an example of a typical *Craigslist* ad, which highlights the applications of the heuristic algorithm described in Algorithm 3. The ad consists of the name of the person with his phone number. When the concept 'Beverly' is considered, we observe that it is preceded by a digit sequence as described in Section 6.1.7, thereby boosting the score for 'Beverly' as the street concept. Also, the distance between city identified as 'Dallas' and the concept 'Beverly' is zero words, which further adds to the score.

Finally, the algorithm determines two POIs, namely SMU (Southern Methodist University) and UT Southwestern (University of Texas South Western), both being academic institutions are present in the TIGER gazetteer. We then calculate the distance between the street in Dallas by the name of 'Beverly' and the POIs, and boost the score

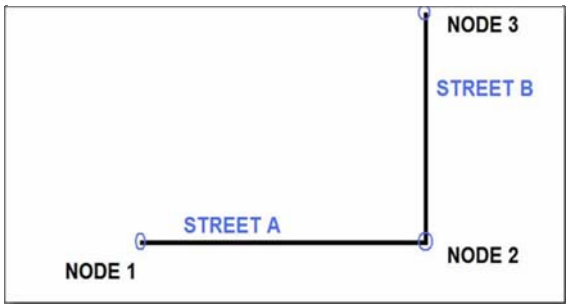
accordingly. Since Beverly is four blocks away from SMU and 1.5 miles from UT Southwestern, the close proximity results in a huge score boost for the 'Beverly' concept.

It is thus important to note here that the high score boost of 'Beverly' facilitates its disambiguation as the correct location concept from other candidates likes 'Jay', Oaklawn, etc. thereby achieving geo-geo and geo/non-geo disambiguation.

6.3 Creating the intersection database

The intersection database is created from the TIGER shape files. TIGER dataset contains the streets divided into segments, each uniquely identified by the starting and the ending Nodes IDs. If two streets intersect, they will have at least one of the two nodes common to both as shown in Figure 4.

Figure 4 The creation of intersection database (see online version for colours)



We perform this pre-processing, check the database for two different named streets having one node in common. We also store the geometric coordinates of the intersection to be able to identify an intersection. Algorithm 4 describes the algorithm to find the intersection from the street-weight vector.

Algorithm 4 Intersect (S, W)

Input: (S, W): Streets and weights vector

Output: (S1, S2, R): Two intersecting streets and R is true for intersection; false otherwise

```
S1 ← Street_max-weight (S, W)
for each S0 ε S
    S2 ← { S0 : S0 ε S, S0 ≠ S1 and |W (S0) – W (S1)|<W_min-diff }
    if TIGER_INTERSECT (S1, S2) == true then
            return (S1, S2, true)
return (null, null, false)
```

In the first step, select the street with the maximum weight as S_1. Then, among the remaining street concepts it looks for a street-concept, S_2 that is different from S_1, but has similar weight and is in close proximity to it. It then checks the intersection database for an intersection. If there is an intersection, it returns true along with the names of the two streets; false otherwise.

7 Confidence-accuracy factor

After we have identified and disambiguated the highest weighing concept from the text, which refers to the street name or city name, we calculate its confidence-accuracy factor (CAF). CAF, a number between 0 and 1, is a measure of the accuracy and the confidence of the apartment's location. Accuracy defines the exactness, or correctness we have of the location, e.g., a street name will have a lower accuracy as compared to street-intersection which will have a lower accuracy as compared to an address with a house number and street name. The confidence part of CAF describes the source of the location. It ascertains the belief in correctness of the source. Hence, an address obtained from the reverse phone number lookup will have a low confidence as compared to a Google/Yahoo maps link. For a location determined from the heuristic-based disambiguation algorithm, the confidence value is based on the final score of the concept. This means if all the heuristics apply to the concept the weight of the concept will be very high and hence we will be more confident of the location identified. Depending on the CAF value, we map the apartment either as a cloud for low, a dart for medium and a house for high confidence-accuracy factor.

$$CAF = CAF_{confidence} + CAF_{accuracy}$$

$$CAF_{confidence} = \sum \alpha_i / (2 * W_{max})$$

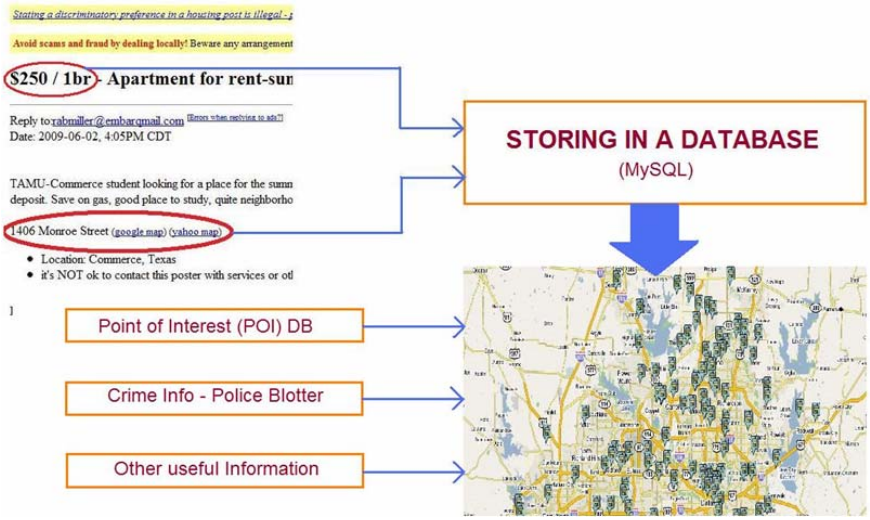
where, α_i is the confidence factor of the source and W_{max} is the maximum confidence (e.g., a Google/Yahoo link with a phone no. which verifies it).

$$A_{google/yahoo} > \alpha_{disambiguation\text{-}algo} > \alpha_{reverse\text{-}phone}$$

$$CAF_{accuracy} = \beta_i / (2 * \beta_{max})$$

where β_i is the accuracy factor and β_{max} is the accuracy value for a location with block number, street and city (most accurate). $B_{block\text{-}street\text{-}city} > \beta_{intersection\text{-}city} > \beta_{street\text{-}city} > \beta_{city}$

Figure 5 Flowchart showing the working of the mash up (see online version for colours)



7.1 Updating TIGER and the uncertainty

The TIGER dataset for the street segment does not contain the city name in which the segment lies. But we can easily obtain this using a reverse geo-coding service, which returns city name in response to the Lat/Long values of the street segment. But on closer observation, it was found that for some street segments different cities were returned by the different web services, e.g., for the Lat/Long value of (33.1166, –96.851), the city name returned by Google Maps is Frisco, whereas that returned by Terra server is The Colony.

As our pruning-heuristic algorithm relies heavily on the accuracy of the city, it becomes necessary that the city name is correct. Thus, there arises a need for data quality, an assessment of the completeness, currency, logical consistency, and accuracy of data. To ensure this we used several reverse geo-coding web services. One way to do this would be using simple majority vote, that is, we choose the city name that is voted by most of the data sources. But the problem with this approach is that it does not take into account the reliability of the data source. Here reliability is defined by the internet traffic and the frequency of updating of the web service. We assigned each data source with a reliability factor (RF) which is a function of the sum of the daily usage and the year the source was last updated in. For the implementation purposes we use three data sources namely, the Microsoft Terra server, Google Maps, and lastly Geonames.org. Table 1 shows the information obtained from Alexa.com.

Table 1 shows information collected from Alexa.com regarding the traffic

	Google Maps	*Terra server*	*Geonames.org*
Daily usage (in millions)	304.80	1.24	10.26
Last updated	Updated regularly	1,998	Updated regularly

Now, consider a case of conflict among N sources, which return M unique cities for a street segment. To resolve the uncertainty, we assign each city instance a weight, which is the sum of all the RFs of the data sources which return that city. Finally, we choose the city which has the maximum weight and update the gazetteer.

8 Integration of data sources

While looking for an apartment, apart from the basic things like rent, location etc. the user is also interested in other facts like the safety of the neighbourhood, the nearby public amenities like parks, schools etc. Figure 5 shows the flowchart describing this integration. From the ad we extract the information like rent, location, number of bedrooms and bathrooms and store it in a relational database. We also have location based information like crime, point of interests stored. In this we describe how we collect, analyse and integrate this information and show it on the map in a way that makes more meaning to the user.

8.1 Points of interest (POI)

Points of interest refer to the various specific point locations that someone may find useful or interesting. We had a comprehensive database of the over 300 POIs, provided

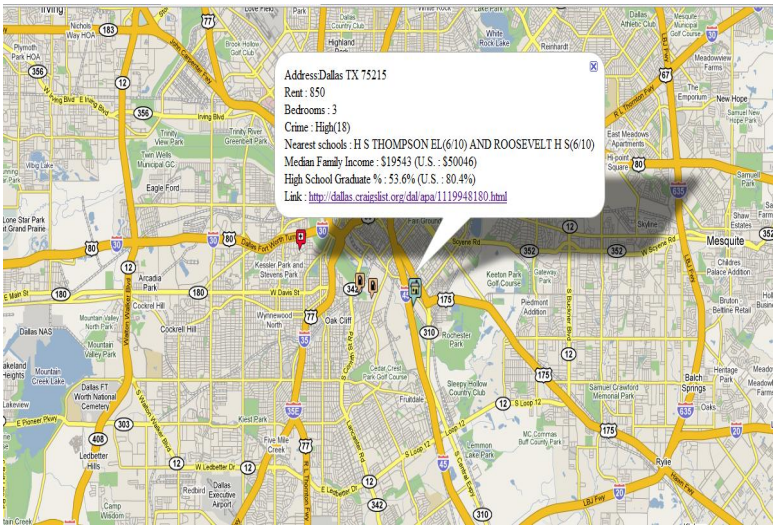
by Homeland Security Information Program (HSIP). Amongst them, we selected ten that pertained to the interests of someone looking for an apartment. These included grocery stores, places of worship, parks, gas stations, schools etc. So when a user is looking an apartment apart from the basic things like rent, no. of bedrooms he also can see the nearest POIs on the same map.

8.2 Crime statistics and other information

Safety of the neighbourhood is a key considering while someone is looking for an apartment. For this reason, we try to provide certain pointers for the type of neighbourhood such as the crime rating, median family income and the percentage of high school graduates. For the crime rating, we periodically scan the police blotter, aggregate the information, and present it in a scale from 0 to 10 so that it makes more sense to the user. Similarly we integrate the median family income and percentage of high school graduates and display it along with other details of the apartment.

Figure 6 shows a potential apartment with nearest gas stations and hospitals. Clicking on the apartment icon gives other information about it including location, rent, bedrooms, crime rating, median family income, high school graduates and the link to the ad on Craigslist.

Figure 6 Screenshot #3 showing a potential apartment with information and nearby hospitals and gas stations (see online version for colours)



9 Results

We used a dataset comprising of 1,000 randomly chosen ads from the Craigslist website for one day for the Dallas Fort Worth listings. We then geo-tagged these first by looking for Google/Yahoo maps link, a direct physical address or reverse phone lookup in them. This portion of the results guarantees 100% accuracy. Figure 7 shows the internal distribution of output.

More than half of these had a map link; rest was almost equally divided between the exact physical address and the reverse phone number lookup. Then, for the remaining we applied our disambiguation algorithm and then manually checked the geo-tags for correctness. The algorithm either returned a location as a neighbourhood or street name or in the absence of a high weight street returned 'null' indicating the absence of a street name.

Figure 7 The internal distribution of ads with physical address (see online version for colours)

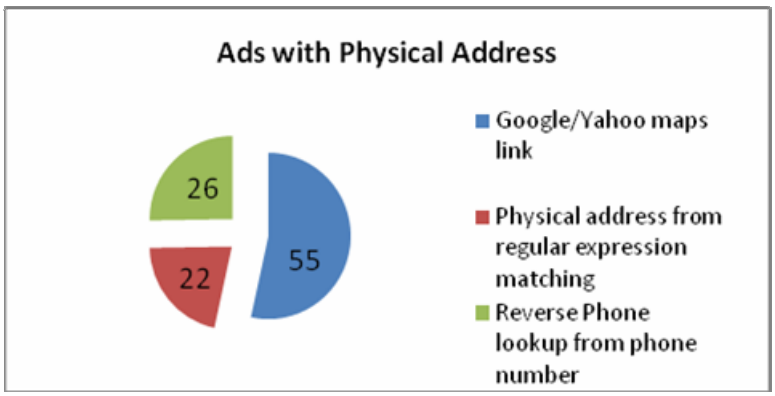
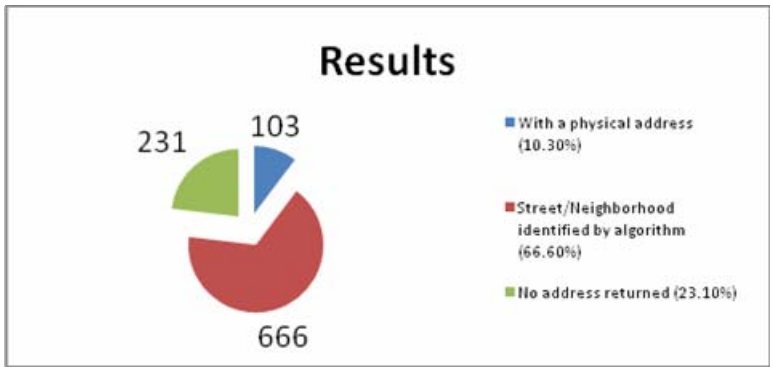


Figure 8 shows the results of our approach. For 103 (10.30%) documents we were able to either get an address from Google/Yahoo map link or physical address from regular expression matching or White pages reverse phone lookup. Next, on this set of 897 we used the algorithm, for which, the algorithm returned street names for 122 and identified 544 neighbourhoods and returned 'null' for the remaining 231 (23.1%).

Figure 8 Results of the Craigslist-Google Maps system (see online version for colours)



We then tested the correctness of our algorithm by manually annotating the entire set of documents and comparing it with the results. Figure 9 shows the Precision, Recall and F-measure values for the dataset using different approaches.

Using our approach, we get a precision of 0.577, recall of 0.776 and an F-measure of 0.662. The other figure shows the precision and recall using a simple gazetteer-based approach optimal for city level disambiguation with no heuristics. The main reason for the higher performance of our algorithm is the identification of the neighbourhoods and

the heuristics. Next, we show the impact of various heuristics on the accuracy. The heuristics are grouped together for simplicity. Firstly we check the accuracy by using the just gazetteer based approach. Next we apply the heuristics of name lookup, reverse phone lookup, city distance, commonly occurring words and traffic count (Heuristics 6.1.1–6.1.5). Finally we add to these heuristics N-gram, digit sequence, street suffix and POI (Heuristics 6.1.6–6.1.9).

Figure 9 Shows the precision and recall values of our system compared to just simple gazetteer-based approach (see online version for colours)

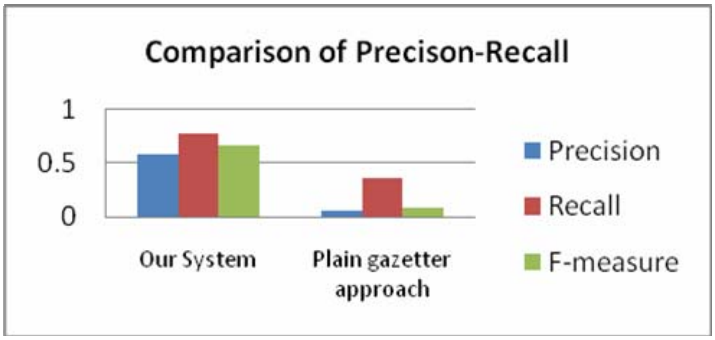
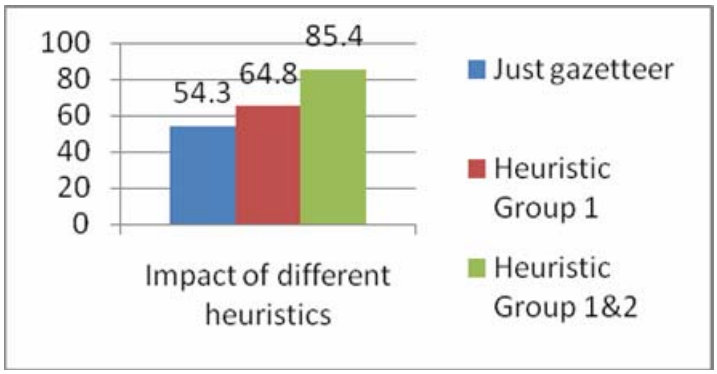


Figure 10 Impact of various heuristics and pruning on the results (see online version for colours)



10 Conclusions and future works

We developed an apartment searching tool, MapIt, which takes the Craigslist ads as the source and shows them on Google Maps, integrated with other services such as crime ratings, points of interest etc. making it easier for the user to come to a decision. We also make use of a disambiguation algorithm to correctly identify the location of the apartment and to increase the coverage. With each apartment we also associate a CAF value to give the user an idea the confidence and accuracy we have in the correct positioning of the location.

The results show a significant increase in the coverage as compared to other sites. Since the data is so unstructured and the annotation of street names is a difficult task,

there is a still a segment of ads for which we still could not find a location. In future we would like to extend our system to increase the coverage by improving the algorithm. Other future work includes improving the GUI, making the system more user-oriented by giving him preferences to narrow down the search, and include user reviews.

References

Abrol, S., Khan, L, (2010) 'Agglomerative clustering on fuzzy k-closest friends with variable depth for location mining', *The Second IEEE International Conference on Social Computing (SocialCom2010)* 20–22 August 2010, Minneapolis, Minnesota.

Abrol, S., Khan, L. and Al-Khateeb, T.M. (2009) 'MapIt: smarter searches using location driven knowledge discovery and mining', *1st SIGSPATIAL ACM GIS 2009 International Workshop on Querying and Mining Uncertain Spatio-Temporal Data (QUeST)*, November 2009, Seattle.

Amitay, E., Har'El, N., Sivan, R. and Soffer, A. (2004) 'Web-a-Where: geo-tagging web content', in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp.273–280.

Hecht, B., Hong, L., Suh, B. and Chi, E.H. (2011) 'Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles', to appear at *ACM CHI'11.*

Li, H., Srihari, R.K., Niu, C. and Li, W. (2002) 'Location normalization for information extraction', in *Proceedings of the 19th International Conference on Computational Linguistics*, August, pp.1–7, Taipei, Taiwan.

Lieberman, M.D., Samet, H., Sankaranarayanan, J. and Sperling, J. (2007) 'STEWARD: architecture of a spatio-textual search engine', in *Proceedings of the 15th International Symposium on Advances in Geographic Information Systems,* ACM GIS 2007.

McCurley, K.S. (2001) 'Geospatial mapping and navigation of the web', in *Proceedings of the 10th Int. Conference on World Wide Web*, pp.221–229, ACM Press.

Shekhar, S. and Chawla, S. (2003) *Spatial Databases – A Tour*, Prentice Hall, Upper Saddle River, New Jersey (ISBN 013-017480-7).

Smith, D.A. and Crane, G. (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), Lecture Notes in Computer Science,* pp.127–136, Darmstadt, September 2001.

Websites

Allurstuff: http://www.allurstuff.com

CRF Tagger: http://sourceforge.net/projects/crftagger/

Housingmaps: http://www.housingmaps.com

Hover's Craigslist-Company Overview.

MapsKreig: http://www.apskreig.com

Padmapper : http://www.padmapper.com

TIGER gazetteer: http://www.census.gov/geo/www/tiger/

ZIP code statistics by US Census Bureau: http://www.census.gov/epcd/www/zipstats.html