

Mining Geospatial Path Data from Natural Language Descriptions

Nate Blaylock

Bradley Swain

James Allen

Institute for Human and Machine Cognition (IHMC)
40 South Alcaniz Street
Pensacola, Florida, USA
{blaylock, bswain, jallen}@ihmc.us

ABSTRACT

In this paper, we describe the TEGUS system for mining geospatial path data from natural language descriptions. TEGUS uses natural language processing, GIS entity databases, and graph-based path finding to predict lat/lon paths based only on natural language text input. We also report on preliminary results from experiments on a corpus of path descriptions.

Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural Language Processing—*Language parsing and understanding*

General Terms

Algorithms

Keywords

Geospatial language understanding, information extraction

1. INTRODUCTION

There exists a wealth of geospatial information in text documents, audio streams, blogs, and so forth that is encoded in qualitative descriptions in natural language. Our ultimate goal is to be able to extract such information from arbitrary text or speech. In this paper, we tackle the problem of *geospatial path understanding*: extracting a path in lat/lon coordinates given a natural language description of that path. Path understanding would enable a number of applications, including automated geotagging of text and speech, robots that can follow human route instructions, and NL-description-based localization.

Our approach to this problem uses natural language understanding to extract references to geospatial entities (such as businesses, streets, intersections, parks, etc.) These linguistic references are used to formulate queries to large GIS

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM QUES^T09 November 3, 2009. Seattle, WA, USA
Copyright 2009 ACM 978-1-60558-853-7 ...\$10.00.

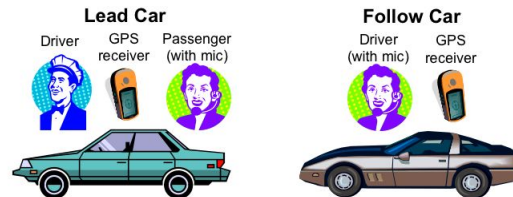


Figure 1: A session in the PURSUIT Corpus

databases to search for possible grounding *referents* (the actual entity that was referred to (e.g., *meant*) in the language). Both reference extraction and search are noisy processes which produce uncertain results. However, we are able to use context to efficiently narrow results to those which form a globally consistent path using a graph search algorithm.

In the remainder of this paper, we first describe the corpus we use for training and testing. We then describe the path understanding system and then our first experimental results using this system. We then discuss related work and conclude by mentioning future directions.

2. THE PURSUIT CORPUS

In order to test and train our algorithms, we created the PURSUIT Corpus [2], which contains path descriptions in natural language, with each location reference annotated with the name and lat/lon coordinates of the real-world referent. In this section we describe the data collected and the annotations on it.

2.1 Data

The PURSUIT Corpus was gathered using a novel method for speech data collection. In it, subjects describe their path in real time (i.e., while they are traveling it) and a GPS receiver simultaneously records the actual path.

These GPS tracks of the actual path can aide the annotator in determining which GIS entities/locations were meant by each geospatial reference. To be clear, the accompanying GPS tracks are used only as “ground truth” data in *evaluating* the path understanding system, which only operates on the linguistic description as input.

Figure 1 shows an example of the experimental setup for the corpus collection. Each session consisted of a lead car and a follow car in downtown Pensacola, Florida. The driver of the lead car was instructed to drive wherever he wanted

for an approximate amount of time (around 15 minutes). The driver of the follow car was instructed to follow the lead car. One person in the lead car (usually a passenger) and one person in the follow car (usually the driver) were given close-speaking headset microphones and instructed to describe, during the ride, where the lead car was going, as if they were speaking to someone in a remote location who was trying to follow the car on a map. The speakers were also instructed to try to be verbose, and that they did not need to restrict themselves to street names—they could use businesses, landmarks, or whatever was natural. Both speakers’ speech was recorded during the session. In addition, a GPS receiver was placed in each car and the GPS track was recorded at a high sampling rate.

The corpus contains 13¹ audio recordings of seven paths along with two corresponding GPS tracks from the cars. The average session length was just over 18 minutes. Some sample utterances from the corpus are given below:

- *...and we’re going under the I-110 overpass I believe and the Civic Center is on the right side on the corner of Alcaniz and East Gregory Street where we are going to be taking a left turn...*
- *... he’s going to turn left right here by the UWF Small Business Development Center heading toward Gulf Power ...*
- *... we’ve stopped at a red light at Tarragona Street okay we’re going now across Tarragona passing the Music House ...*
- *... we’re at the intersection of East Gregory and 9th near a restaurant called Carrabas I think and a Shell station just a little south of the railway crossing ...*

2.2 Annotation

The speech data was synchronized with the GPS tracks, then transcribed and segmented into utterances by hand. The GPS tracks provide ground truth for the path described, but not for references to geospatial entities used in describing the path. In order to better test the path understanding system, we also hand annotated the classes of geospatial entities available in our GIS databases (description below), using a custom tool for annotating and synchronized replay of speech and GPS track information in Google Earth [3]. The high-level classes annotated were:

- *Streets*: references to a given street, for example “Garden Street” or “a divided road”
- *Intersections*: references to street intersections, for example “the corner of 9th and Cervantes” or “the next intersection”
- *Addresses*: references to street address, for example “401 East Chase Street” or even “712” (when referring to the address by just the street number)
- *Other Locations*: this class is a grab bag for all other location types that we annotated, consisting of such data as businesses, parks, bridges, bodies of water, etc.

¹In one session only one audio recording was made.

| | Named | Category | Total |
|--------------|--------------|--------------|-------------|
| Street | 77.2% | 22.8% | 48.5% |
| Intersection | 45.5% | 54.5% | 6.8% |
| Address | 100.0% | 0.0% | 0.8% |
| Other Loc | 67.7% | 32.3% | 43.9% |
| Total | 71.1% | 28.9% | 100% |

Table 1: Breakdown of geospatial entity reference annotations in the PURSUIT Corpus

Note that not *all* geospatial entity references have been annotated in PURSUIT—just those that are accessible in our GIS databases. Examples of entities that appear in the corpus but were not annotated are fields, parking lots, sidewalks, railroad tracks, and fire hydrants. These were not annotated only because we did not have access to data about those entities. However, there is nothing inherent in our approach to path understanding which would prohibit the use of those classes of entities, if data were available for them.

Although not all *classes* of entities were annotated, within those classes that were annotated, *all* references to entities of interest were annotated, whether or not they were named. Thus “Garden Street”, “a divided road”, or even “it” were annotated if they referred to a geospatial entity of interest.

Annotations are also marked with whether an entity reference was *named* (i.e., contained at least part of the proper name of the entity, such as “the Music House” and “the intersection at Cervantes”) or *category* (description did not include a name, such as “the street”, “a Mexican restaurant”, and “it”).

All entity references of interest were minimally annotated with their canonical name and a lat/lon coordinate. Streets were annotated with the lat/lon of the street segment from the database closest to the speaker’s current location. Where applicable, entities were also annotated with a street address. In cases where the entity was not in the databases, the human annotator searched for the missing data by hand using various resources.

In total, 1649 geospatial entity references were annotated in the corpus. The breakdown of categories is shown in Table 1.

3. THE PATH UNDERSTANDING SYSTEM

In this section, we describe our path understanding system TEGUS (The Geospatial language Understanding System). The general architecture of TEGUS is shown in Figure 2. Input to the system is the natural language text description of a path. The description is parsed to a Logical Form (LF), from which references to geospatial entities are extracted. These references are used to generate queries to two GIS databases, producing a list of possible entity matches. The list of references and possible matches are then used in a path finding algorithm which uses temporal and geospatial constraints to predict the most likely path.

Note that, in the current form of the system, the path that is recognized is not based on street segments. Rather, the assumption is made that geospatial entities being mentioned will be in the close vicinity of the current position of the car. The “path” predicted by the system is actually an ordered list of geospatial entities and their lat/lon coordinates. In Section 6, we discuss our plans to move to a street segment

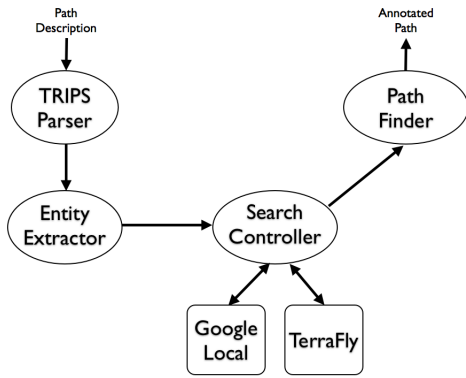


Figure 2: The TEGUS System Architecture

based system.

It is also important to note that, although GPS track information was gathered as part of the corpus, it is *not* used as input to the path understanding system. The only input to TEGUS is the text description of the path in natural language. (The GPS information is used for evaluation, however, as described below.)

We now describe each of the system’s subcomponents.

3.1 Language Processing

Language processing is done by the TRIPS Parser system [1], which is the language understanding component of the TRIPS dialogue system [5]. TRIPS performs deep syntactic/semantic analysis on the text using a hybrid symbolic-statistical model. The result of analysis is a graphical Logical Form (LF). Several external components are used to guide the symbolic parser, which helps ensure broad coverage. These include a statistical shallow parser, several large lexical resources including WordNet, and the Stanford Named Entity Recognizer (NER) [4]. During processing, output from these components are used as *suggestions* to the TRIPS parser. However, the parser takes a larger context into account and it is free to heed or ignore these suggestions. (A detailed description of the TRIPS Parser system is outside the scope of this paper. However, we mention the Stanford NER especially, as it is used to customize the parser in our experiments described below.)

The Logical Form representation for “He’s coming down Spring” is shown in Figure 3.

3.2 Location Reference Extraction

LFs for the utterances are then passed to the Entity Extractor. This component uses hand-built semantic graph matching rules to find the subgraphs of the LF that correspond to location references. In the case for the LF shown in Figure 3, the entity extractor matches on the **GEOGRAPHIC-REGION** subtree and returns it.

3.3 GIS Search

TEGUS currently accesses two GIS databases, although the architecture will allow any number of databases that may be available. The first database, TerraFly [10], is a large aggregation of GIS data. Relevant parts of TerraFly used in TEGUS are a street segment dataset with NavTeq

data; a custom-built dataset of street intersections based on the street segment database; and a large, geographic points dataset with businesses, bridges, parks, schools, restaurants, hotels, bodies of water, etc. aggregated from several datasets including: GNIS, NAVTEQ POI, Yellow Pages, and US Census data. The second database is Google Local², which provides API-level access to a large database of businesses and points of interest.

Both databases support queries for keywords and a lat/lon for the center of the search. Currently, we make the simplifying assumption that the start and end points of the path are the same, and that we know the lat/lon coordinate for that point. We also assume that the path stays within 3 miles of the starting location. Below we discuss plans to relax these assumptions in future work.

The Search Controller converts each extracted location reference into one or more queries to the GIS databases above. Except for cases where the LF clearly marks the high-level class of the entity (e.g., such as knowing that the referent is a street), queries are executed to all of the databases. Search results are aggregated these become the list of possible referents for the given location reference.

3.4 Path Finding

The search controller may return many possible referents for a location reference. The Path Finder component uses global context choose the best referent for each location reference. This is done by constructing a directed graph of all possible referents for all location references in the session and finding the optimal path between the start and end points. Although we formulate the Path Finder as a graph search algorithm, it is important to note that geometric distance is not the only factor in determining edge weights in our graph. We are not literally trying to find the shortest path (in physical sense), but rather the best fit for the given linguistic description.

The search graph is constructed with each possible referent as a node, and edges connecting all nodes from one location reference to the next. This simple graph forms a trellis, and the optimal path through the trellis visits exactly one node for each location reference, which we can then use to predict the referent for that reference. Unfortunately, a trellis alone will not work for this problem. The assumption that each layer contains the correct referent is too rigid, as it is possible that *none* of the entities in the list is the correct answer for the location reference. This can (and does) happen because of problems at any stage of processing. For example, a non-location may be incorrectly extracted, or the GIS databases may not contain the actual referent. Whatever the reason, we need to be able to handle the case where no node is the correct answer for a given location reference.

To do this, we add additional edges from a node to all nodes up to N references ahead of it. This allows the optimal path algorithm to skip a location reference (thus not making a prediction about that node). Although this multiplies the number of edges in the graph, the constant “skip” factor (N) keeps this number manageable. Optimal paths through the graphs produced in our experiments were speedily found. The experimental results below were attained by allowing the algorithm to skip up to 10 references (i.e., $N = 10$).

Next in the graph construction process, TEGUS uses temporal and geometric knowledge to delete any edges that

²<http://maps.google.com>

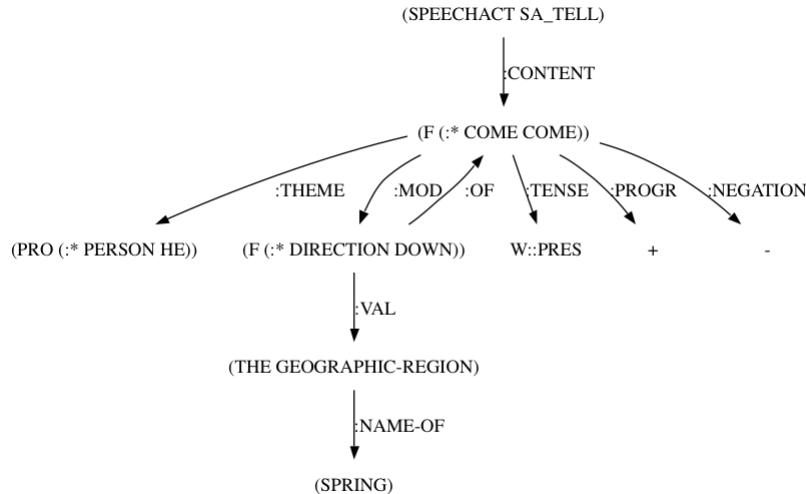


Figure 3: Logical Form for “He’s coming down Spring”

are considered impossible transitions. Because the corpus is based on speech, we are able to preserve the timing information from between utterances. With this, we know how much time has transpired between two location references. Using this, together with knowledge about limitations on the mode of transportation (a car), TEGUS can determine whether the (physical) distance between two possible referents was possible given a maximum assumed car speed (100 km/h). Remaining edges are weighted by the distance between the two entities plus a penalty for each location reference “skipped” by the edge.

The remaining edges are assigned as weight based, preliminarily, on physical distance as well as name similarity. The latter is currently a simple average of similar word tokens between the two entities. As discussed below, we believe more sophisticated similarity measures will help increase accuracy.

Once the graph is constructed, Dijkstra’s Algorithm is used to compute the least cost path through the graph. This path represents TEGUS’ prediction of the actual path taken.

4. EXPERIMENTAL RESULTS

We performed a preliminary evaluation of the TEGUS system on the PURSUIT Corpus. To evaluate, we performed a 7-fold cross validation (one for each session) of TEGUS on the PURSUIT Corpus. The Stanford NER was trained on all *named* location references from the remaining 6 sessions and used with the TRIPS parser. Each path description was separately input to the TEGUS system, which produced a single path prediction.

4.1 Evaluation

We evaluate the results of this experiment in two ways, using the GPS track and hand-annotations of the corpus. The first evaluation is on the path prediction itself. As noted above, the current version of the system does path prediction by location references as opposed to street segments. Except for street mentions, very rarely will the speaker have been exactly on the lat/lon coordinates of the geospatial entities he mentions. However, our assumption, as explained above,

is that these entities will often be very close to the speaker. We check correctness of the predicted path by looking at the distance between the predicted location and the position of the car (as given by the GPS track) at the time of the location is referred to in speech. If the distance is within 300m, it is counted as a correct prediction.³

As the algorithm counts on location mentions to predict current location, it is unreasonable to expect that the system will make predictions at the same rate the GPS location was sensed (e.g., 1 Hz). Instead, we take as the maximum number of expected prediction the number location references (from the hand-annotated corpus). This number is used in computing recall for the path prediction.

Besides the final path result, we also evaluate the system’s performance on correctly resolving each location reference to the corresponding referent (the actual geospatial entity). Here, we also measure precision and recall. Also, similar to what we do with the path, we allow some distance between the predicted and actual points. As we are working with different databases, because of the inaccuracies of geocoding or other errors, entries for the same entity in different databases can have different names and/or lat/lon coordinates. Because of this, entities predictions within 200m of those in the corpus are counted as correct predictions. This number was chosen anecdotally from examples where the same address was geocoded to different lat/lons in Google Local and TerraFly.

Results on the test corpus for *path prediction* were **92.6% precision** and **49.2% recall**. Results on geospatial entity *reference resolution* were **88.9% precision** and **47.2% recall**.

Reference resolution statistics broken down by reference and entity type are shown in Table 2.

4.2 Discussion

We first note that these results are somewhat preliminary,

³The allowable distance from the path will ultimately depend on the end application using geospatial path understanding.

| | Named | | Category | | Total | |
|---------------------|--------------|--------------|--------------|-------------|--------------|--------------|
| | Prec. | Recall | Prec. | Recall | Prec. | Recall |
| Street | 96.8% | 73.9% | 100.0% | 1.1% | 96.8% | 57.3% |
| Intersection | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Address | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| Other Loc | 86.2% | 60.0% | 43.5% | 11.5% | 79.7% | 44.3% |
| Total | 92.4% | 64.0% | 45.3% | 6.1% | 88.9% | 47.2% |

Table 2: Breakdown of reference resolution results by entity and reference type

and we expect to improve both precision and recall through the future work described below. All-in-all, however the results seem very good, especially the precision in path prediction. Recall numbers are only in the 40’s, although for most extraction tasks, precision is far more important than recall. Many of the missed cases in recall are actually quite difficult, if not impossible to do based on the current model of location-based path prediction. References such as “the road” or “this intersection” hold very little informational content, making search for them nearly impossible (without returning *all* the street segments or intersections in the search radius). We believe that moving to a street-based modeling will lead to much higher rates of recall.

In reference resolution, resolution for street references was quite high with 96.8% precision, and *other loc* prediction has 88.9% precision. TEGUS does not yet try to extract intersection or address references (which is why no predictions were made), but we include them for completeness.

The performance on named versus category references is quite stark. Overall, for named references (references with at least one proper name in them), TEGUS achieved 92.4% precision and 64.0% recall. For category references (references with no proper names), however, the system only achieved 45.3% accuracy and 6.1% recall. This is attributable to several factors, including the “the road” type references with little informational content mentioned above, and the fact that category references (such as “a bank”) usually match a much wider range of entities than a named reference (such as “Wells Fargo”) does.

It is also interesting to note that the precision result for path prediction is almost 5% higher than precision on reference resolution. What seems to be happening here is that, in some cases, the algorithm is predicting an entity which is *not* the intended referent, but nonetheless is still close to the car position. This is partly due to the fact that entities may have similar names in similar areas (e.g., *Alcaniz Street* and *Atelier Alcaniz* (which is on Alcaniz Street)). Category-based descriptions are sometimes also clumped (e.g., many banks or restaurants in the same vicinity). Additionally, Google Local often returns results which not only match the keywords from the search, but also entities that are in the general vicinity from those entities. In these cases, the path finding algorithm is faced with a choice among several entities that are very close to each other. We believe that weighting edges based on a better name similarity metric will help here.

As mentioned above, there are several simplifying assumptions that undoubtedly help the results. The assumption that we know the start and end locations of the path is unrealistic in most cases. In future work below, we discuss plans to remove this assumption.

5. RELATED WORK

Although there has been a fair amount of work on geospatial reference extraction and resolution ([6, 7, 8], *inter alia*), most has been at the level of cities, provinces, and countries. Our work focuses on the sub-city level and the resolution of references to streets, intersections, and businesses. The sub-city contains a much larger set of possible referents, and databases are much less complete than they are for information about cities and states.

As far as we are aware, very little work has been done in the area of mining geospatial path data from descriptions. The work in [9] tries to extract paths from text descriptions, although it, too, is at a city and province level as opposed to a sub-city level of granularity.

6. CONCLUSION AND FUTURE WORK

In this paper, we have presented the TEGUS system for path understanding. TEGUS uses natural language processing, access to large GIS databases, and graph-based path finding to mine geospatial paths from natural language descriptions. We also presented preliminary experimental results on the PURSUIT corpus.

There are several areas of future work. First, in the short term, there are many incremental improvements that can be made to the system. As mentioned above, currently edge weight in the graph is based on physical distance and a simple name similarity metric (and skip penalties), which is not a good model of the problem. Consistent graphs should be treated equally, not whether one is geometrically shorter than the other. Instead, we plan to weight edges based (at least partly) on more sophisticated similarity metrics between the original search keywords and the actual name and category of the entity found, such as described in [11].

There are also several simplifying assumptions which we will try to remove from the system. First is the assumption that the start and end point of the path are the same and are known to the system. The second is that we know the radius of the search. We will replace these assumptions with only the assumption that we are operating in a given region (defined by a bounding box), and that the start and end location can be in arbitrary points within that bounding box. It may be reasonable to assume that one knows the general area that the path was in. Our ultimate goal, of course, is to start without any a priori knowledge of location and still find the path. Doing so efficiently may be a challenge.

Additionally, although temporal information can be used when processing speech, it is not available for text sources (e.g., blogs). We will also try to run the system without the assumption that we have temporal information from spoken utterances. Currently temporal information is used to make certain transitions in the path impossible based on a set

maximum speed. Not knowing *when* the references were made should make this a more difficult problem.

In the longer term, we plan to move from the current location-based path prediction to a model which tracks the current street segment the car is on, and uses a wider range of natural language understanding to understand not only location references, but also movement (e.g., “we just turned left”) and orientation (e.g., “we’re going south”) information. This should result not only in increased precision and recall, but also increased frequency with which the system can predict the user’s current position.

7. REFERENCES

- [1] J. F. Allen, M. Swift, and W. de Beaumont. Deep semantic analysis of text. In *Symposium on Semantics in Systems for Text Processing (STEP 2008) Shared Task: Comparing Semantic Representations*, Venice, Italy, September 22-24 2008.
- [2] N. Blaylock and J. Allen. Real-time path descriptions grounded with GPS tracks: a preliminary report. In *LREC Workshop on Methodologies and Resources for Processing Spatial Language*, pages 25–27, Marrakech, Morocco, May 31 2008.
- [3] N. Blaylock, B. Swain, and J. Allen. TESLA: A tool for annotating geospatial language corpora. In *Proceedings North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT) 2009 Conference*, Boulder, Colorado, May 31–June 5 2009.
- [4] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of ACL*, Ann Arbor, Michigan, June 2005.
- [5] H. Jung, J. Allen, L. Galescu, N. Chambers, M. Swift, and W. Taysom. Utilizing natural language for one-shot task learning. *Journal of Logic and Computation*, 18(3):475–493, 2008.
- [6] J. L. Leidner, G. Sinclair, and B. Webber. Grounding spatial named entities for information extraction and question answering. In *Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References*, Edmonton, Alberta, 2003.
- [7] I. Mani, J. Hitzeman, J. Richer, D. Harris, R. Quimby, and B. Wellner. SpatialML: Annotation scheme, corpora, and tools. In *6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, May 2008.
- [8] A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazetteers. In *Proceedings of EACL '99*, pages 1–8, Bergen, Norway, 1999.
- [9] J. Pustejovsky and J. L. Moszkowicz. Integrating motion predicate classes with spatial and temporal annotations. In *Proceedings of the COLING 2008*, Manchester, UK, 2008.
- [10] N. Rishe, M. Gutierrez, A. Selivonenko, and S. Graham. TerraFly: A tool for visualizing and dispensing geospatial data. *Imaging Notes*, 20(2):22–23, 2005.
- [11] O. Uryupina. Evaluating name-matching for coreference resolution. In *4th International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, May 2004.