Geospatial Route Extraction from Texts

Euthymios Drymonas
Institute for the Management of
Information Systems/RC ATHENA
G.Bakou 17, 11524, Athens, Greece
+30 2106990522
edrimon@imis.athena-innovation.gr

Dieter Pfoser
Institute for the Management of
Information Systems/RC ATHENA
G.Bakou 17, 11524, Athens, Greece
+30 2106990522
pfoser@imis.athena-innovation.gr

ABSTRACT

The need to collect vast amounts of geospatial data is driven by the emergence of geo-enabled Web applications and the suitability of geospatial data in general to organize information. Given that geospatial data collection and aggregation is a resource intensive task typically left to professionals, we, in this work, advocate the use of information extraction (IE) techniques to derive meaningful geospatial data from plain texts. Initially focusing on travel information, the extracted data can be visualized as routes derived from narratives. As a side effect, the processed text is annotated by this route, which can be seen as an improved geocoding effort. Experimentation shows the adequacy and accuracy of the proposed approach by comparing extracted routes to respective map data.

Categories and Subject Descriptors

H.3.1 [INFORMATION STORAGE AND RETRIEVAL]: Content Analysis and Indexing - Indexing methods, Linguistic processing

General Terms

Algorithms

Keywords

user contributed geospatial data, data extraction, information extraction, geospatial data, natural language processing

1. INTRODUCTION

Geographic information, be it maps or 3D virtual worlds, are believed to be the future way for people to socialize, shop, and share information. In the foreseeable future, the map will become the interface of choice for the Internet [23]. In an increasing number of (Web) applications space is however not only used as metadata to structure and access information, but also as the actual content resource. Overall, the most significant advantages of geospatial data are its (i) unambiguous nature, i.e., categories and keywords are up for interpretation, a geographic coordinate is not, and (ii) the simplicity of the matching interface, i.e., maps.

To "geo enable" the Web, two major issues need to be addressed. One, content needs to be related to geographic co-ordinates, i.e., geocoded. Second, sufficient amounts of geospatial data need to be available, e.g., (road) networks, address information, POIs, routes, etc. *Geocoding* has been exhaustively addressed not only in literature but also by a series of products (cf. Google Maps API [11]), all sharing the basic approach of comparing text strings to gazetteer entries that are linked to coordinates. A different issue is the availability of *sufficient geospatial data sets* for any types of application. Here, with the proliferation of the Internet as the primary medium for data publishing and information exchange, we have seen an explosion in the amount of online content available on the Web. Thus, in addition to professionally-produced material being offered free on the Internet, the public has also been allowed, indeed encouraged, making its content available online to everyone by means of *user-contributed content*. The aim is to harness the ability humans have to massively collect and share knowledge with the ultimate goal of digitizing the world (from a geospatial point of view). As early maps were traces of people's movements in the world, i.e., view representations of people's experiences, digitizing the world in this context relates to collecting pieces of knowledge gained by a human individual tied not only to space and time, but also to her context, personal cognition, and experience. Through *intentional* (e.g., narratives, geo-wikis, geocoding photos) or *unintentional effort* (e.g., routes from their daily commutes), simple users create vast amounts of data concerning the real world that contain significant amounts of information. The ambitious aim in such a crowdsourcing effort will be however to go beyond purposefully contributed data and to *include any type of available content such as existing Web pages in the data collection effort*. This potentially vast amount of data will lead to a digitized world beyond mere collections of co-ordinates and maps.

Of importance to both, the geocoding and the crowdsourcing approach, is an *understanding of textual content with respect to the geospatial data that it contains*. To this respect, this work proposes an Information Extraction (IE) approach based natural

ACM SIGSPATIAL DMG Workshop, Nov 2, San Jose, CA, U.S.A.
Copyright 2010 ACM ISBN 978-1-4503-0430-6/10/11...$10.00."

language processing (NLP) techniques towards the understanding, detection and extraction of geospatial data nuggets from texts. Using text engineering methods, we propose for the context of travel information a (extendable) set of rules that allows us to detect travel information in written texts. This rule base can be extended; however, our experimentation showed that after considering a certain number of documents an accurate detection of route information in newly added texts can be achieved. We combined this approach with geocoding and routing functionality to derive actual route information. As such the proposed approach is a hybrid system incorporating, both, aspects of geocoding, i.e., texts are not only related to location information but to actual routes, and geospatial data extraction, i.e., actual route information is extracted from texts without having any prior knowledge. An empirical evaluation using actual texts from travel guides and travel diaries shows the usefulness and accuracy of the proposed approach.

Related work exists towards two general directions, (i) geocoding and (ii) extraction of routes from texts. With respect to geocoding, we can exemplary cite [16], one of the first works on geocoding and describing a navigational tool for browsing web resources by geographic proximity as an alternative means for Web navigation. Web-a-Where [1] is another system for geocoding Web pages. It assigns to each page a geographic *focus* — a locality that the page discusses as a whole. The tagging process targets large collections of Web pages to facilitate a variety of location-based applications and data analyses. The approach presented in [3] proposes the use of the Web's geographic information to populate address databases, i.e., parse Web pages for useful address information and populate an address database with the available information. The work presented in [13] is identifying and disambiguating references to geographic locations. A method for calculating the geographic breadth of a Web page is given in [9]. Another method that uses information extraction techniques to geocode news is described in [22]. In the realm of geocoding, a range of related *commercial products* exist. Google Maps API provides geocoding services [11]. A similar service is Yahoo Yellow Pages [17]. What is common to those services is that they simple try to geocode a given input string. MetaCarta on the other hand [15] provides tools and services that also geoparse and then geocode text content using natural language processes and highly refined geodata. The approach in [12] uses state-of-the-art tools in their work for extracting geographical information from data. The results can be used for geographic search on the Web, in GIS applications, for categorizing documents, etc. However, no evidence is given to the existence of a route extraction mechanism. In this context, we can cite [18], which aims at mapping natural language descriptions to a custom-created sidewalk database, i.e., this approach is not generally applicable to arbitrary routes since developed in a controlled environment and limited vocabulary. Work towards the classification of route-relevant expressions is presented in [26]. However, no actual routes are produced. [7] aims at extracting a transportation network graphs from Web documents. Using a given set of seed locations, Web documents are retrieved to identify candidate transportation nodes between the locations.

The outline of the remainder of this work is as follows. The contribution, namely geospatial data extraction from texts using information extraction methods is detailed in Section 2. Based on this approach, Section 3 presents an experimental evaluation that focuses on route extraction from texts. Finally, Section 4 gives conclusions and directions for future research.

2. GEOSPATIAL DATA EXTRACTION FROM TEXTS

In our approach, we apply Information Extraction (IE) methods for deriving geospatial content from narratives. IE is generally defined as the process of locating user-specific information in electronic documents, "*the name given to any process which selectively structures and combines data which is found, explicitly stated or implied, in one or more texts"* [5]. In the present effort our focus will be on content that contains rich geospatial data such as travel literature.

Given travel guides and travel diaries, our objective is to correctly recognize location and direction information so as to construct actual route datasets that can be visualized on a map.

2.1 Overview

A precursor to extracting route information from texts and to actually construct a map, is to extract a meaningful and coherent series of points that describe the narrated route.

There is a huge amount in the WWW of texts regarding spatial content, like travel blogs or travel diaries and guides that contain rich information that could be exploited and organized in automatic ways. Instead, apart from the fact that users contribute their own narratives every day, these documents are not analyzed by computers in order to exploit semantic information and are treated as bags of words. Our method makes use of state-of-the-art Information Extraction methods to derive meaningful information by analyzing such free text narratives, extracting names of places as well as relative information between them. In this work we extract information like "head north for 20 meters and meet Key bar". We extract relative and absolute information regarding a place. This information would reveal places that cannot be geocoded (for example "Key bar" that a geocoder can't recognize), but mentioned explicitly in a text narrative. Thus a main advantage of our method is that we use only linguistic, semantic and contextual information contained in free text narratives, without making use of supervised methods (e.g., gazetteers, lists) in order to extract meaningful named entities (i.e. places) and relations between them.

By using IE techniques, we also try to bypass the important problem of *ambiguity*, i.e., not falsely linking identifiers to coordinates such as when the name of a geographic location shares a non-geographic meaning as well (George Washington vs. Washington DC) or distinct geographic locations share the same name (London, England vs. London, Ontario). Disambiguation of geographic entities is achieved by properly identifying the context of the identifier in a sentence. In addition, IE techniques help in addressing the problem of incomplete gazetteers and place name variations and abbreviations.

The IE system used in this work consists of three principal parts, (i) the linguistic pre-processing part, (ii) the document IE semantic analysis part (the core feature extraction process) and

(iii) the geocoding part. The various system components and relationships are shown in Fig. 1. The system has been implemented as a pipeline application of individual tools using the GATE - General Architecture for Text Engineering platform [6], a software framework for natural language processing and engineering. GATE allows for the embedding of different types of language resources (ontologies, lexicons, etc.) and modules that perform various types of processing in the form of plugins (CREOLE components). Each component has to be implemented as a Java Bean with a well defined input/output interface. Furthermore GATE provides a convenient graphical interface for developing and/or evaluating components for various natural processing tasks (cf. the use of this interface in visualizing annotations in Section 3). For a specific processing task, an arbitrary number of components may be used sequentially in what is termed a processing *pipeline*.

In what follows, we describe in necessary detail the processing pipeline, which overall uses a document in plain text as input and, as shown in Section 3, produces a map in the form of a KML file [19] that can be viewed by means of, e.g., Google Earth.

2.2 Linguistic pre-processing tools

Linguistic pre-processing tools analyze natural language documents in terms of words, sentences, part-of-speech and morphology. We selected the ANNIE tools, contained in the GATE release, to perform this initial part of analysis. To this task, our processing pipeline comprises of a set of four modules: (i) the ANNIE tokeniser, (i) the (ANNIE) Sentence Splitter, (iii) the ANNIE POS Tagger and (iv) the WordNet Lemmatiser.

The intermediate processing results are passed on to each subsequent analysis tool as GATE document annotation objects. The output of this analysis part is the analyzed document in CAS/XML format, an XML scheme called Common Annotation Scheme allowing for a wide range of annotations, structural, lexical, semantic and conceptual [21]. This document is temporarily stored in the system, so as to be accessed by the subsequent CAFETIERE semantic analysis component. CAFETIERE combines the linguistic information acquired by the pre-processing stage of analysis with knowledge resources information, namely the lookup ontology and the analysis rules to semantically analyse the documents and recognize spatial information.

The first step in the pipeline process is *tokenisation,* i.e., recognising in the input text basic text units (tokens), such as words and punctuation and *orthographic analysis*, i.e., the association of orthographic features, such as capitalisation, use of special characters and symbols, etc. to the recognised tokens [25]. The tools used are ANNIE Tokeniser and Orthographic Analyser. The ANNIE tokenizer distinguishes five types of tokens: *word, number, symbol, punctuation* and *space* tokens. The orthographic analysis process of the tool is paired with tokenisation analysis rule-based processing and distinguishes four orthographic categories for the respective token types: *upperInitial, allCaps, lowercase* and *mixedCaps* categories. These token types will be used in the rule-based CAFETIERE IE engine for recognizing placenames and spatial relations.

Sentence splitting, in our case the ANNIE sentence splitter aims at the identification of sentence boundaries in a text. Though a seemingly trivial task, sentence splitting can become quite complex due to the ambiguous or dual function of certain punctuation marks. A dot, for example, may indicate both an abbreviation and a sentence end and, among other uses, it can also be employed in acronyms and as indicator of decimal digits of a real number.

Part-of-speech (POS) tagging is the process of assigning a part-of-speech class, such as Noun, Verb etc. to each word in the input text. The ANNIE POS Tagger implementation is a variant of Brill Transformation-based learning tagger, which applies a combination of lexicon information and transformation rules for the correct POS classification.

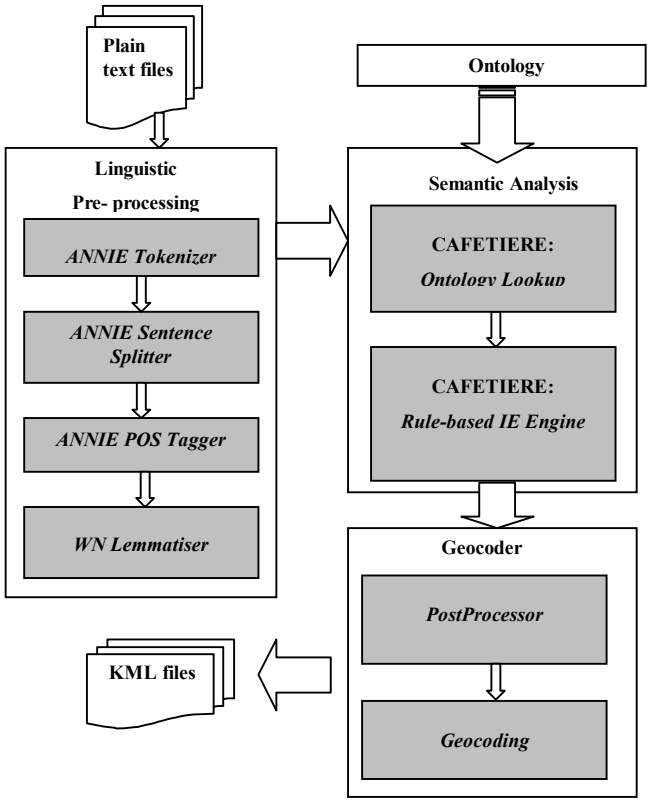


Fig. 1. GATE pipeline

Lemmatisation is used for text normalisation purposes. With this process we retrieve the tokens' base form e.g., for words: ["travelling", "traveler", "traveled"], "are", the corresponding lemmas are: "travel", "be". We exploit this information in the semantic rules section. For this purpose we implement the JWNL WordNet Java Library API [8] for accessing the WordNet relational dictionary. The output of this step is included it in GATE document annotation information.

2.3 Semantic Analysis Tools

Semantic analysis relates the linguistic processing results to ontology information and applies analysis rules, i.e., *documents*

are analyzed semantically to discover spatial concepts and relations.

CAFETIERE [2] is a standalone system combining linguistic pre-processing and relevant for our work, semantic analysis. The CAFETIERE Information Extraction Engine module objective is to compile the set of the semantic analysis grammar rules in a cascade of finite state transducers so as to recognise in text the concepts of interest. For this purpose the CAFETIERE IE Engine combines all previously acquired linguistic and semantic (lookup) information with contextual information. We modified CAFETIERE to process documents in a GATE pipeline and perform only ontology lookup and rule-based semantic analysis. The input to this process are the GATE annotation objects resulting from the linguistic pre-processing stage stored in CAS/XML format for each individual document.

2.3.1 Cafetiere Ontology Lookup

The CAFETIERE Ontology lookup module accesses a previously built ontology to retrieve potential semantic class information for individual tokens or phrases. All types of conceptual information, related to domain specific entities, such as terms or words in general that denote spatial concepts or properties and relations of domain interest are pre-defined in this ontology, built by an expert. The ontology used in our experimentation was created by manually analyzing a large number of texts and iteratively refining the ontology so as to automatically produce results that are close to what an expert user would have manually created.

Consider the partial ontology shown in Fig. 2. Class "LOCVERB" stores verbs that when matched to a text phrase are likely to indicate a spatial relationship between the corresponding referenced concepts. For example, in the phrase "*cross over the bridge and head to Fifth Avenue*", the existence of words contained in "LOCVERB" class denoting spatial information, like "*cross*" and "*head to*", help us derive the desired information. Approximating the human notion when building phrases, we are extracting "Fifth Avenue" as a desired place name from this sample sentence.

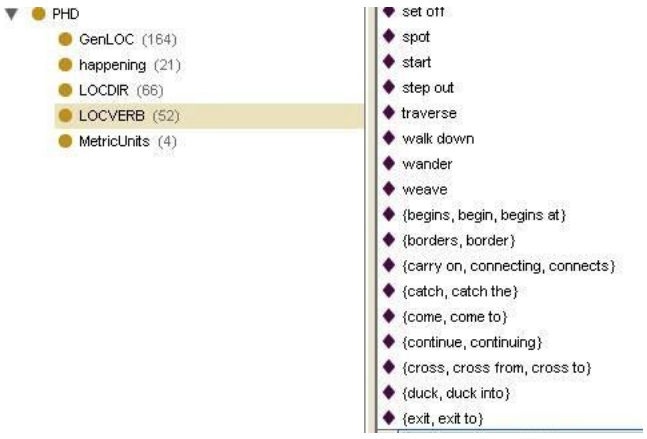


Fig. 2. Sample ontology contents (Protégé ontology editor)

With the application of semantic rules that make use more characteristics of the language, like part-of-speech or orthography (e.g., named entities are written uppercase), as we will see in Section 2.3.2, we are extracting the wanted features from text. Also, the results of this process do not include at this stage any information regarding place names extraction. The subsequent application of the semantic analysis rules undertakes the tasks of disambiguation and the extraction of spatial information. The lookup ontology consists of OWL statements. We can easily add or remove semantic classes or their respective instances by using an ontology editor such as Protégé [20] as shown and used in the example of Fig. 2.

2.3.2 Cafetiere Rule-Based IE engine

The semantic analysis rules, based on CAFETIERE specifications are developed as a single set of context-sensitive/context-free grammar (CSG/CFG) rules.

The CAFETIERE Information Extraction Engine module objective is to compile the set of the semantic analysis grammar rules in a cascade of finite state transducers so as to recognise the concepts of interest in plain texts. For this purpose the CAFETIERE IE Engine combines all previously acquired linguistic and semantic (lookup) information with contextual information found in the plain texts. The semantic analysis rules, are developed as a set of context-sensitive/context-free grammar (CSG/CFG) rules.

An example of a CAFETIERE rule formalism is as follows:

```
[s=__x, target=__trglabel,
rulid=relation8]=>
\
[lookup="LOCVERB", pos=VB, token=__x],
[lookup="LOCDIRECTION", token=__x],
[pos=IN, token=__x]?,
[pos=DT, token=__x]?,
[orth=uppercase, token=__trglabel,
token=__x]{1,4},
[lookup="GenLOC",
tokentoken=__trglabel, token=__x]?
/
```

In this rule formalism, the left part of the rule, before the arrow symbol (=>) is called left-part side of the rule (LHS), while the part appearing after the arrow symbol is called right-hand side of the rule (RHS). Each constituent of the RHS is in the form of single minimal textual units where words in the sentence are matched, while the LHS describes features where the final extracted text spans will be held. In our specific sample rule, LHS contains the rule's id and two features, *s* and *target*, where we store the final information. For the above sample rule to be applied, the sentence snippet that should be matched should start with a verb matched in the lookup ontology as a verb denoting spatial information, the immediate next token should be a word showing directional information (ex. north, south), followed by a token with a part-of-speech tag of *IN* or *DT* (i.e. *preposition/subordinating conjunction* or *determiner*, as defined in [14][1]. The rule formalism provides both standard iteration (?,

[1] Example site with penn tagset: http://www.mozart-oz.org/mogul/doc/lager/brill-tagger/penn.html

+, *) and iteration range operators (e.x. in the above rule {1,5} means 1 to 5 times of consecutive uppercase tokens). The output placename will be written in the LHS entity reference feature named *target*.

As an example text snippet, let us consider the following example: "*From the tower, head east along the Amstel river to take in the ...*". The rule above specifies a pattern where firstly a token (i.e., "*head*"), matching the ontology class "LOCVERB" is extracted as an instance of the respective class, and denoting a verb that could be expressing spatial information. This token is recognized by the POS tagger as verb, so it also matches the required rule POS feature "VB". In the same way, the other tokens are recognized, with respect to their POS tag or their appearance in the lookup ontology. For example the POS tags *"in"* (preposition/subordinating conjunction) and *DT* (determiner) are matching the tokens "*along*" and "*the*", respectively. Finally, a token with an orthography typical for proper names (i.e., *uppercase*) is matched and since it co-occurs in a sentence with the other rule constituents, it is recognized as a spatial object.

In conclusion, the incremental variable *__trglabel,* attached in the *target* feature of the rule gets the value "*Amstel river*". By incremental variable we mean that the matched tokens after each one matched rule constituent are kept into this variable. Similarly, we capture in the *s* feature the contents of incremental variable *__x*, which is the phrase "*head east along Amstel river*". For more information about the CAFETIERE rule formalism, the reader is referred to [2]. The phrase "*Amstel river*" will be kept for geocoding by the Geocoding pipeline module, while the phrase "*head east along Amstel river*" kept in the *s* feature will be annotated visually in the GATE platform by the PostProcessor pipeline module (cf. Fig. 3).

The output of CAFETIERE is stored in a CAS/XML file, which for this example, is as follows:

```
<tok id="t211" pos="VB" lem="head"
   lookup="LOCVERB"
   orth="lowercase">head</tok>
<tok id="t212" pos="JJ" lem="east"
   lookup="LOCDIR" orth="lowercase"
   >east</tok>
<tok id="t213" pos="IN" lem="along"
   lookup="NIL"
   orth="lowercase">along</tok>
<tok id="t214" pos="DT" lem="the"
   lookup="NIL"
   orth="lowercase">the</tok>
<tok id="t215" pos="NNP" lem="amstel"
   lookup="NIL"
   orth="upperInitial">Amstel</tok>
<tok id="t216" pos="NN" lem="river"
   lookup="GenLOC"
   orth="lowercase">river</tok>
<tok id="t217" pos="TO" lem="to"
   lookup="NIL"
   orth="lowercase">to</tok>
<tok id="t218" pos="VB" lem="take"
   lookup="LOCVERB"
   orth="lowercase">take</tok>
<tok id="t219" pos="IN" lem="in"
   lookup="NIL"
   orth="lowercase">in</tok>
<Prelation id="pr2"
   label="head east along amstel
   river"
   source="" target="Amstel river"
   rulid="relation8" tokrefs="t211
   t212 t213 t215 t216" />
```

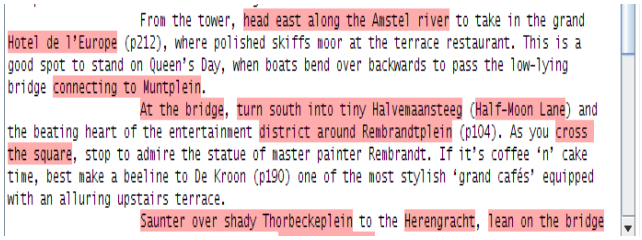
Note that ids like "t211" were assigned by GATE to each token in the previous pre-processing step and they are kept in feature "*tokrefs*".

The rules are stored in a plain text file, which is read during the initialization of the CAFETIERE module, thus allowing us to easily provide our system with new rules.

In the following sections, we describe the two modules that follow the semantic analysis process, namely the Postprocessor and the Geocoding module.

2.4 Postprocessor

The Postprocessor collects the output results of semantic analysis from the CAS/XML and relates it to the original text. Using the Castor tool [4], this module passes the token ids back to GATE to create annotation sets for the actual documents examined. Fig. 3 shows such a sample annotation for walk descriptions in a travel guide (cf. content used in experiments of Section 3). The Postprocessor module then passes the results (like "Amstel River" from the previous example) to the Geocoder.


From the tower, head east along the Amstel river to take in the grand Hotel de l'Europe (p212), where polished skiffs moor at the terrace restaurant. This is a good spot to stand on Queen's Day, when boats bend over backwards to pass the low-lying bridge connecting to Muntplein.

At the bridge, turn south into tiny Halvemaansteeg (Half-Moon Lane) and the beating heart of the entertainment district around Rembrandtplein (p104). As you cross the square, stop to admire the statue of master painter Rembrandt. If it's coffee 'n' cake time, best make a beeline to De Kroon (p190) one of the most stylish 'grand cafés' equipped with an alluring upstairs terrace.

Saunter over shady Thorbeckeplein to the Herengracht, lean on the bridge

Fig. 3. Original document annotated with extracted content

2.5 Geocoding and Routes

The semantic analysis provided us (among others) with *place name information*, i.e., the place name identifiers contained in the text, e.g., Amstel river, Muntplein, etc. To determine their actual location, these identifiers need to be geocoded. For this task we rely on the open-source module GeoGoogle [10], a Java API utilizing the Google Geocoder service, which is part of the Google Maps API [11].

The retrieved results are of varying accuracy. In the experiments of Section 3, only results of GeoAddressAccuracy >= 5 (cf. [11]) were used. This value corresponds to "street level accuracy", i.e.,

somewhere on a specific road. In addition, spatial outliers are detected by calculating the distance between sets of points, i.e., if a retrieved geocoding result would extend a path by more than *x* km it is omitted. For the case of the city guides used in Section 3, *x* was chosen to be 1km.

To then retrieve a *route*, the filtered geocoded place marks need to be connected so as to create a valid road path. In order to tackle this problem, we implemented a Java wrapper for directions feature of the Google Maps API (i.e., a wrapper similar to GeoGoogle for routing). This wrapper allows us to compute a shortest-path between place marks using the Google street network data. The result comprises a polyline for the route to follow in the road network. This is the final step of our pipeline implementation, with the geocoder module creating respective KML files of the respective routes. The following section showcases this approach and gives actual routes from example datasets.

2.6 Summary

Focussing on texts that contain route information, we use an information extraction approach that utilizes a location ontology to describe spatial relationships and properties in combination with a rule-based IE engine to extract place and, connecting them in sequence, route information. A main advantage of our system is that we *do not rely on exhaustive gazetteer lists*, but a *relatively small in size ontology to annotate texts and extract geospatial data*.

The following section gives some specific examples that show the applicability of the approach.

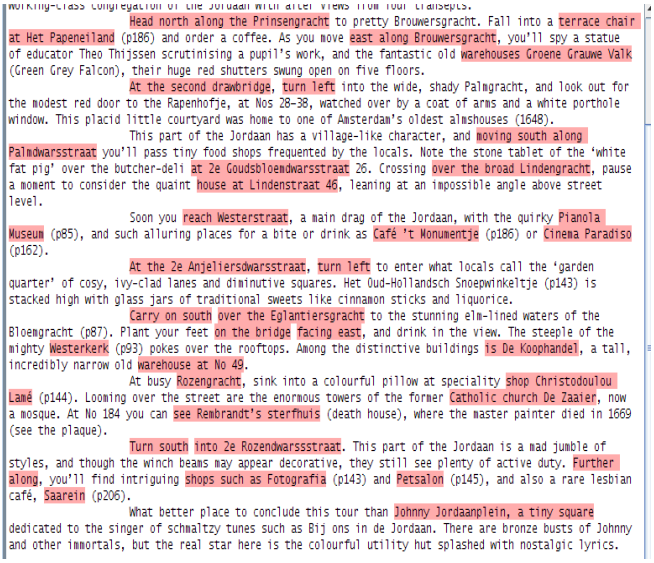
3. Experimental Evaluation

The following experimental evaluation tries to assess the quality of the proposed approach by comparing textual route descriptions with their actual map counterparts. For this purpose, we used content from actual Lonely Planet travel guides (Amsterdam, Budapest and Melbourne). In those guides walking tours are given by means of (i) a textual description and (ii) an accompanying map. Our objective was to recreate the map by processing the textual description with the approach advocated in Section 2, i.e., extract place information and geocoded as many as possible to actual show the created route on a map. The results are given in the following figures, which show a Google Earth visualization of the resulting KML next to the original travel guide map (Fig. 5).

3.1 Travel Guides and Routes

A complete route extraction example is shown in Fig. 4 (© Lonely Planet, Amsterdam City Guide – content used under "Fair Use" terms). Fig. 4(a) shows the annotated text of the guide after being processed by the IE system. Placemarks and movement information is highlighted. Fig. 4(b) visualizes the route extracted from the annotated text after being processed by the geocoder and routing engine. When compared to the original route that accompanied the text in Fig. 4(b), the two routes, although not an exact match are very similar.

Table 1 gives an overview of the actual text sizes and annotation results of the various city guides used in this experimentation and the respective processing results. For example, the text as shown partially in Fig. 4(a) comprises 520 words and 38 phrases were annotated, i.e., marked as containing place names or other relevant spatial information. Out of those annotations, 25 were actual place names and using GeoGoogle, we were able to geocode 10 entries. The resulting route is shown in Fig. 4(c). Respective numbers are given for the other three case studies. It is worth mentioning that *the quality of the resulting route highly depends on the geocoding tool as in the Amsterdam example, only 10 out of 25 recognized place names were geocoded.* Nevertheless, the produced result resembles the original route to a very large degree.



working-class congregation of the Jordaan with artist views from four transepts.

Head north along the Prinsengracht to pretty Brouwersgracht. Fall into a terrace chair at Het Papeneiland (p186) and order a coffee. As you move east along Brouwersgracht, you'll spy a statue of educator Theo Thijssen scrutinising a pupil's work, and the fantastic old warehouses Groene Grauwe Valk (Green Grey Falcon), their huge red shutters swung open on five floors.

At the second drawbridge, turn left into the wide, shady Palmgracht, and look out for the modest red door to the Rapenhofje, at Nos 28–38, watched over by a coat of arms and a white porthole window. This placid little courtyard was home to one of Amsterdam's oldest almshouses (1648).

This part of the Jordaan has a village-like character, and moving south along Palmdwarsstraat you'll pass tiny food shops frequented by the locals. Note the stone tablet of the 'white fat pig' over the butcher-deli at 2e Goudsbloemdwarsstraat 26. Crossing over the broad Lindengracht, pause a moment to consider the quaint house at Lindenstraat 46, leaning at an impossible angle above street level.

Soon you reach Westerstraat, a main drag of the Jordaan, with the quirky Pianola Museum (p85), and such alluring places for a bite or drink as Café 't Monumentje (p186) or Cinema Paradiso (p162).

At the 2e Anjeliersdwarsstraat, turn left to enter what locals call the 'garden quarter' of cosy, ivy-clad lanes and diminutive squares. Het Oud-Hollandsch Snoepwinkeltje (p143) is stacked high with glass jars of traditional sweets like cinnamon sticks and liquorice.

Carry on south over the Eglantiersgracht to the stunning elm-lined waters of the Bloemgracht (p87). Plant your feet on the bridge facing east, and drink in the view. The steeple of the mighty Westerkerk (p93) pokes over the rooftops. Among the distinctive buildings is De Koophandel, a tall, incredibly narrow old warehouse at No 49.

At busy Rozengracht, sink into a colourful pillow at speciality shop Christodoulou Lamé (p144). Looming over the street are the enormous towers of the former Catholic church De Zaaier, now a mosque. At No 184 you can see Rembrandt's sterfhuis (death house), where the master painter died in 1669 (see the plaque).

Turn south into 2e Rozendwarssstraat. This part of the Jordaan is a mad jumble of styles, and though the winch beams may appear decorative, they still see plenty of active duty. Further along, you'll find intriguing shops such as Fotografia (p143) and Petsalon (p145), and also a rare lesbian café, Saarein (p206).

What better place to conclude this tour than Johnny Jordaanplein, a tiny square dedicated to the singer of schmaltzy tunes such as Bij ons in de Jordaan. There are bronze busts of Johnny and other immortals, but the real star here is the colourful utility hut splashed with nostalgic lyrics.

(a)

(b)

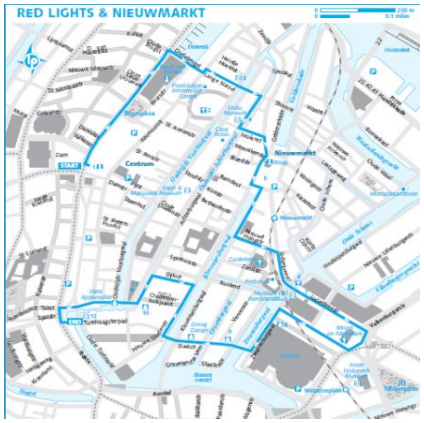
(c)

Fig. 4. Amsterdam – route extraction example

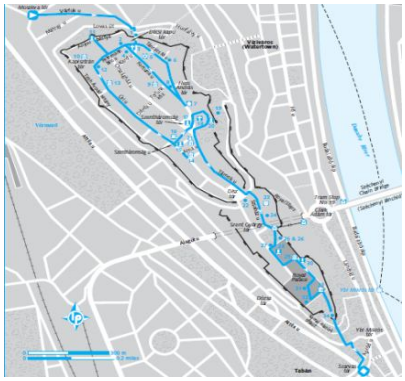
Further route extraction examples are shown in Fig. 5, including another route for Amsterdam, plus routes for Budapest, Hungary and Melbourne, Australia. Although in each case not all place names identified in the text were geocoded, each route clearly resembles the original one shown by means of a respective map. With better geocoding algorithms, which are beyond the scope of this work, the obtained route results could be considerably improved.

Table 1. Texts and processing results

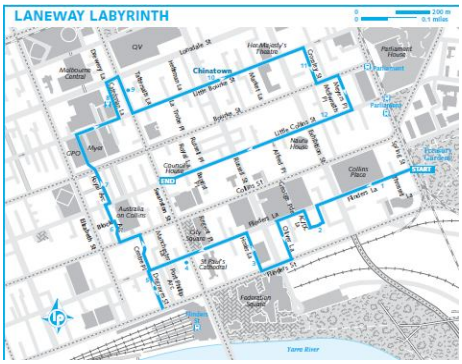
	Amsterdam/ Jordaan	Amsterdam – Nieuwmarkt	Budapest	Melbourne
Nof. words total	520	566	1625	647
Annotations found	38	42	101	47
Place names found	25	32	58	35
Geocoded Place names	10	16	21	18
Geocoding percentage	*40%*	*50%*	*36%*	*51%*



(a) Amsterdam



(b) Budapest



(c) Melbourne

Fig. 5. Route extraction examples (© Lonely Planet, City Guides)

3.2 Narratives and Routes

Fig. 6 shows a somewhat different example in the form of a travel diary containing narrative about a trip to Benin, Africa. The extracted route information and KML visualization are shown respectively. Please note that while more text portions have been identified, geocoding failed due to a lack of gazetteer data. This example should illustrate that our proposed approach is universally applicable and produces results for various types of content.

Activities Benin is a perfect destination for those seeking a facinating glimpse into a complex culture. As far as organised tourist programs are concerned, though, it's pretty light-on. For a low-key safari experience, head to the far-north to Pendjari Park and W Park. Pendjari is more developed for tourists than W, and is only open between mid-December and mid-May. The park contains elephants, hippos, buffalo and lions, but you'll be lucky if you see more than a few forlorn-looking antelope, a couple of wart hogs and maybe a monkey or three. The coastline is spectacular, and especially well suited to swimming. Four km (2.5mi) east from the centre of Cotonou is the best urban beach. It's safe, clean and regularly crowded. Head west for 40km (25mi) and you'll find the absolutely perfect beach at Ouidah (which just happens to be the voodoo capital of Benin). Just a little better than perfect, and a mere 40km (25mi) further along the road, is Grand Popo. The beaches are quite safe, the sand is a spectacular white and the water clear and clean.

Fig. 6. Travel diary example.

4. CONCLUSIONS AND FUTURE WORK

Extracting geospatial data from texts is becoming a pressing need considering the data requirements posed by emerging Web applications utilizing geospatial data. Not wanting to rely on professional data creators, because of financial, data coverage, accuracy, etc. reasons, we will have to define tools that will allow anybody to contribute to a global geospatial data stash. This work contributes an information extraction system that (i) extracts routes from texts and (ii) goes beyond simple geocoding by actually annotating texts with routes. A main advantage of our system is that we provide plain narrative texts and we do not rely on exhaustive gazetteer lists, but a relatively small in size ontology to annotate texts and extract geospatial data. The approach is based on natural language processing techniques that provide robustness and also accuracy. Our system extracts not only route information but actual contexts of spatial objects as identified in texts. The experiments show that the proposed approach is suitable for extracting with considerably accuracy actual routes from narrative and, thus, creating geospatial data and increasing the value of the provided content.

Directions for future work are as follows. Although not examined in depth in this work, the context of spatial objects such as spatial (spatiotemporal) relationships (moving from X to Y) is identified in our proposed approach. Hence, the next step will be to map spatial relationships such as metric, topological and directional and their spatiotemporal equivalents to English language expressions and extract such data from texts (cf. [24]). A consequence of this approach will be the creation of a robust rule base for extraction of such relationships. The eventual goal of this work will be to derive arbitrary datasets such as maps automatically from texts. Here, one will have to deal with the uncertainty of user-contributed datasets and respective data fusions techniques.

5. REFERENCES

[1] Amitay, E., Har'EL, N., Sivan, R., and Soffer, A. 2004. Web-a-Where: Geotagging Web Content. In Proc. of SIGIR, 273-280.

[2] Black, W.J., McNaught, J., Vasilakopoulos, A., Zervanou, K., Theodoulidis, B. and Rinaldi, F. 2005. CAFETIERE: Conceptual Annotations for Facts, Events, Terms, Individual Entities and Relations. *Technical Report TR-U4.3.1*, January 11, University of Manchester.

[3] Borges, K. A. V. , Laender, A. H. F., Medeiros, C.B., and Davis, C.A. 2003. The Web as a Data Source for Spatial Databases. In *Proc. 4th ACM Workshop on Geographical information retrieval*, 31-36.

[4] Caster project. Open Source data binding framework for Java. http://www.castor.org. Project page.

[5] Cowie, J. and Wilks, Y. 2000. Information Extraction. In: R. Dale, H. Moisl and H. Somers (eds.) *Handbook of Natural Language Processing*, Marcel Dekker, New York.

[6] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proc. 40th Anniversary Meeting of the Association for Computational Linguistics* (ACL'02).

[7] Davidov, D. and Rappoport, A. 2009. Geo-mining: discovery of road and transport networks using directional patterns. In *Proc. 2009 Conference on Empirical Methods in Natural Language Processing*, Vol. 1, 267-275.

[8] Didion, J. Java WordNet Library (JWNL). http://sourceforge.net/projects/jwordnet. Sourceforge.net project page.

[9] Ding, J., Gravano, L., and Shivakumar, N. 2000. Computing Geographical Scopes of Web Resources. *In Proc. 26th VLDB conference*, 545-556.

[10] GeoGoogle. Google Geocoder Java API. http://geo-google.sourceforge.net/. Sourceforge project page.

[11] Google Inc. Google Maps API. http://code.google.com/apis/maps/. Web page.

[12] Hassan, A., Jones, R., and Diaz, F. 2009. A case study of using geographic cues to predict query news intent. In *Proc. 17th ACM GIS conference*, 33-41.

[13] Lieberman, M.D., Samet, H., and Sankaranarayanan, J. 2010. Geotagging with local lexicons to build indexes for textually-specified spatial data. In *Proc. 26th ICDE conference*, 201-212.

[14] Marcus, M.P., Santorini, B., and Marcinkiewicz, M. A. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2): 313-330.

[15] MetaCarta Inc. Company homepage. http://www.metacarta.com/, Web page.

[16] McCurley, K. 2001. Geospatial mapping and navigation of the web. In *Proc. 10th WWW conf*., 221-229.

[17] Yahoo Inc. Yahoo Yellow Pages. http://yp.yahoo.com/. Web page.

[18] Noaki, K. and Arikawa, M. 2005. A Geocoding method for natural route descriptions using sidewalk network databases," In *Proc. Web and Wireless Geographical Information Systems*, 38-50.

[19] OpenGIS® KML Encoding Standard (OGC KML). http://www.opengeospatial.org/standards/kml/. Web page.

[20] Protégé project homepage: http://protege.stanford.edu. Web page.

[21] Rinaldi, F., Dowdall, J., Hess, M., Ellman, J., Zarri, G. P., Persidis, A., Bernard, L., and Karanikas, H. 2003. Multilayer annotations in Parmenides. In *Proc. KCAP Workshop on Knowledge Mark up and Semantic Annotation*.

[22] Teitler, B. E., Lieberman, M.D., Panozzo, D., Sankaranarayanan, J., Samet, H., and Sperling, J. 2008. NewsStand: a new view on news. In *Proc. 16th ACM GIS conference*, 144-153.

[23] Waters, R. 2008. Way to go? Mapping looks to be the Web's next big thing. *Financial Times*, May 22, 2008.

[24] Xu, J. and Mark, D.M. 2007. Natural Language Understanding of Spatial Relations Between Linear Geographic Objects. *Spatial Cognition & Computation: An Interdisciplinary Journal*, 7(4):311-347.

[25] Zervanou, K. 2007. TOWL Deliverable 5.1 – Design of text feature extraction, version 1.0, October 2007. TOWL: Time-determined ontology based information system for real time stock market analysis, Technical University of Crete.

[26] Zhang, X., Mitra, P., Xu, S., Jaiswal, A.R. and Maceachren, A. 2009. Extracting Route Directions from Web Pages. In *Proc. Int'l Workshop on Web and Databases* (WebDB).