Geographical Information Recognition and Visualisation in Texts Written in Various Languages

Bruno Pouliquen, Ralf Steinberger, Camelia Ignat, Tom De Groeve

European Commission - Joint Research Centre
Institute for the Protection and Security of the Citizen
T.P. 267, 21020 Ispra (VA), Italy
(+39) 0332 78 9135

{Bruno.Pouliquen, Ralf.Steinberger, Camelia.Ignat, Tom.de-Groeve} @jrc.it

ABSTRACT

In this paper, we describe a system that recognises place names in natural language text and produces geographic maps and animations showing the geographical coverage of texts about a certain subject as it changes over time. As the system is built to analyse texts in many different languages, it restricts the usage of linguistic analysis tools to the minimum. Instead, it relies on a gazetteer containing place names in different languages and uses heuristics for disambiguation purposes.

Categories and Subject Descriptors

H.3.1 [**Information Systems**]: Information Storage and retrieval– *content analysis and Indexing.*

General Terms

Algorithms, Design, Experimentation

Keywords

place name recognition; natural language processing; data mining; GIS; named entity recognition.

1 INTRODUCTION

Semantic annotation of texts for further use in information retrieval and information management in general is a raising need. Furthermore, the ability to work with many different languages is compulsory for international organisations and others.

The geographical information found in texts has a special importance when dealing with documents distributed all over the world. According to Gey [8], "a major need is to provide geographic and proper name recognition across languages". Developing a tool for many languages raises the problem of the language-dependent input required.

The *United Nations Geographic Information Working Group* formulated the importance of geographical information in information management very well: "Geographic information is a vital component of the United Nations Information Technology policy. UN organisations recognise the power of 'place' to improve knowledge and decision-making by extending the traditional role of maps to support the rapid integration, analysis and modelling of information critical to achieve improved operational readiness and responsiveness"[1]

In this paper, we present a system that recognises geographical information in multilingual natural language texts and that uses this information for the visualisation of either a single text or of a whole set of texts (e.g. all texts about a specific subject). In the latter case, we also present the change of geographical coverage over time. Additionally, we use this information to provide cross-lingual information access to textual information and to identify texts talking about the same events across languages.

After an overview of the state of the art (section 2), we summarise our approach (section 3) and then focus on the recognition of place names in texts, including the disambiguation procedure (section 4). Section 5 describes the visualisation of the extracted information. In section 6, we present some evaluation. In section 7, we draw some conclusions and point to future work.

2 STATE OF THE ART

Geographical place recognition is a sub-category of named entity recognition (NER). For an introduction to the state of the art of the field of NER, see [4].

Mikheev et al. [12] tried to avoid the use of gazetteers for named entity recognition and ended up concluding that location name extraction without gazetteer (i.e. relying only on linguistic patterns) raises bad results (precision/recall respectively are 46%/59%). Place names differ in this from other named entities because other named entities can be identified better without using gazetteers ("names of locations have fewer reliable contextual clues"). Furthermore, their experiment indicated that a small gazetteer (containing "very common names") performs almost as well as a full gazetteer (91%/92% for precision/recall). According to our own experience, the usefulness of larger or smaller gazetteers is text-dependent.

One of the biggest difficulties in geographical place recognition concerns the disambiguation between several places with the same name (e.g. 'Victoria', the capital of Hong Kong and a village in

SAC '04, Month 14-17, 2004, Nicosia, Cyprus.

[1] Report of the 3rd UNGIWG Plenary Meeting, hosted by the World Bank, Washington DC, USA. 17-19/06/2002. http://www.ungiwg.org

England, among almost 200 others), between place names and common words of a language (e.g. 'And' in Iran) and between place names and person names (e.g. 'Victoria'). However, in the specific NER field of geographical places, a disambiguation process can take into account at least two more parameters: the hierarchy of places (e.g. 'Victoria' is part of 'Hong Kong' and thus part of 'China') and the geographical distance between two places ("from Victoria to Macao…").

For a list of open source gazetteers of place names, see [8]. Most place name lists are monolingual (e.g. containing only 'Venice'), some of them contain also the local name variant (e.g. 'Venezia' in Italian), and only few contain exonyms[2] in various languages (e.g. 'Venedig' in German and 'Venise' in French).

GIPSY [16] is one of the first systems that combines place name recognition and visualisation, but it works only with English texts and it focuses mainly on the visualisation part.

The NAACL 03 workshop on Analysis of geographic references compiled a set of interesting articles on the subject[3], which include:

Bilhaut et al. [2] describe a nice linguistic tool that analyses natural language texts in French in order to recognise complex "spatial expressions" (like "the south of a Bordeaux-Geneva line"). Their aim is to provide a spatial query interface. The tool functionality seems to be impressive, but relies on a strong linguistic analysis to produce a semantic representation of the texts.

Leidner et al. [10] describe a system that is very similar to ours (but just for English). It associates spatial named entities to texts (what they call 'grounding') for displaying them on a map. The system relies on a named entity recognition tool. They present also the use of such a system for question answering. The visualisation part of their work differs from ours because they provide a map of one article at a time. The results only seem to be pleasant to the eye when the article refers to places in a single specific area.

Two other applications are similar to ours in term of visualisation: GeoNode and Informedia:

- GeoNode: [9] "Geospatial News On Demand Environment" is an initiative developed at MITRE (USA). It extracts locations (with other Named Entities) from news articles (including video), the articles are then clustered in *topics*. The interface for visualisation uses the locations to produce maps, or animation of a topic evolution on time. They are currently working with English and Spanish languages.
- InforMedia [3] is a library containing video of broadcast news, their tool extracts place names from the news using speech recognition output, or even overlaid text. Then a user can search the news through spatial queries (i.e. retrieve news for a given place name, country or area). They rely on a Named Entity recognition tool, that works currently for English, Spanish and Serbo-Croatian. They can then animated geographical maps, synchronised with the video playback.

According to Friburger & Maurel [6], 43.9% of proper names are locations and proper names represent about 10% of the words in journalistic text. Gey [8] furthermore says that 30% of content-bearing words are proper names. They both showed that, respectively, clustering and Cross Language Information Retrieval can be improved when considering proper names. GeoNode [9] is using exclusively Named Entities for clustering. Software to identify proper names is therefore an important part of a language technology tool set. For some question answering applications it seems even compulsory to have an effective geo-coding tool (see [2]).

As far as we know, none of the existing systems has been developed for many languages. They almost always rely on linguistic tools (which are not so easily adaptable to new languages[4]). Also the visualisation part of the geo-coding results is often quite weak. With our approach, we are able to visualise geographical information interactively, for single texts or whole document collections, written in one or in several different languages.

3 OUR APPROACH

Our aim, as a first application, is to recognise countries and places in newspaper articles. Given a text that can be written in various languages (we are currently working with more than 20 languages), we want the system to add meta-data to the text, including the countries the text refers to and the cities that are mentioned. We plan to use our tool for several languages (multi-monolinguality), but also to use it as a cross-lingual repository, meaning that the contents of texts in different languages can be compared to each other, or that the contents of one language can be presented in another.

As a container for cities, we currently only use countries, but we store the information regarding 'administrative units' (counties, provinces, etc.) for future further improvements of the algorithm.

The recognition of geographical information, what is called 'geo-coding' [5], is just a means for further uses. It is not a goal as such, so we want to develop the visualisation part of it. The first application will be to visualise the information extracted from one or more documents in a geographical map. The second application aims at showing the evolution of geographical coverage over time.

As we want to use our tools on various languages, they are mainly based on statistics and heuristics, with a minimal linguistic input and with no part-of-speech tagging or grammar. We believe that adding linguistic rules would improve the results, but developing these rules for so many languages is out of our reach (see section 2). We do not expect our tool to perform as well as mono-lingual ones. Without linguistic analysis, it is obvious that the tool will never be able to recognise references to locations such as 'the bigger harbours of Northern Europe'. However, our evaluation showed that the tool performs rather well for the recognition of cities and countries in international texts.

[2] Proper name used in a given language to design a geographical object in a foreign country (in [14]).

[3] All papers of this NAACL workshop are listed at: http://www.metacarta.com/kornai/NAACL/WS9/paper.html

[4] but it is still feasible, see the description of the process in [13].

Table 1: Foreign language names (exonyms) for 'London' in various languages (source: KNAB database).

London (en)		
Lundun (ar)	Lan¡t¡an (ml)	Lontoo (fi)
Llundain (cy)	Lańd¡an (ne)	Lunnainn (gd)
Londres(es/fr/pt)	Lańdan (pa)	Lundúnaborg(is)
Londain (ga)	Londër (sq)	Rondon (ja)
Lākana (haw)	Lańd¡an (te)	**Лондон** (ru)
Londra(it/tr)	Londonas (lt)	Lan¡t¡an (ta)
Lańd¡an (kn)	Londýn (cs)	Luân-đôn (vi)
Londona (lv)	**Λονδίνο(ν)** (el)	eLandani (zu)
Londinium (la)	Landanmyo(my)	Londen (nl)

We currently take as input the 'Global Discovery gazetteer'[5], but are beginning to merge it with the various sources compiled by the KNAB project[6] of the Institute of the Estonian Language. Four important concepts must be in the database: the place name, its spelling variants for various languages, its relative 'importance' (a size information value from "1" = capital of a country, "2"= major city, up to "6" small village/place), its geographical co-ordinates (latitude/longitude) and the country it belongs to. An example of linguistic variations of place name is given in table 1.

Such gazetteers are becoming more easily accessible as many countries are providing free place name lists in order to normalise the denomination of their cities. Significantly, the United Nations has a Group of Experts on Geographical Names. This organisation aims at helping the normalisation of geographical place names in different countries spelled in different languages[7].

In addition to the place name list, we added lists of country ISO codes, of currency names, of adjectives pointing to the country and of names of the people of the country. This means that a hit for the country is generated even if only its currency or its people (e.g. 'Iraqi') are mentioned.

As shown in Table 1, we also have to recognise places in languages that do not use the Latin script, which raises the problem of encoding (character set). We want to store, in the same database, different diacritic characters and even different alphabets and writing systems (Latin, Central-European, Cyrillic and Greek, in the future possibly also Arabic, Chinese, Korean and Japanese…). Therefore, we have chosen UTF-8 as the default encoding for the dictionary. Technically, we are developing tools in object-oriented PERL (since its version 5.8, PERL uses UTF-8 as internal encoding). The gazetteer is stored in an Oracle relational database. The user interface is based on usual web technologies (Apache and CGI).

[5] From Europa Technologies Ltd, http://europa-tech.com/

[6] http://www.eki.ee/knab/knab.htm

[7] http://unstats.un.org/unsd/geoinfo/ungegn.htm

4 RECOGNITION OF PLACE NAMES

The recognition of geographical information in text can be divided into the two sub-processes *geo-parsing* and *geo-coding* [5], where the former refers to the recognition of place names and the latter to the disambiguation and marking-up process.

4.1 Geo-Parser

Our first tool aims at analysing a natural language text and at recognising 'potential' place names. As we are analysing various languages, we must know which language a text is written in. For this purpose, we use an n-gram-based language guesser. The parsing must take into account the language. 'Monaco', for example, is non-ambiguous in English, but in Italian it can also be used to refer to 'München' (Munich) in Germany. In our system, each place name will be recognised if written in either the text language or in the local language of the place name, but not in any other language. A German text will refer to the city of Brussels either by using the German name 'Brüssel' or one of the two local names 'Bruxelles' (French) or 'Brussel' (Flemish), but not by using other variants such as 'Brussels' (English).

When trying to recognise place names in natural language text, we also encounter the problem of declensions and other suffixes. Particularly when dealing with Slavonic and Finno-Ugric languages (Hungarian, Finnish, Estonian), we could not ignore the morphological variants. In Finnish, for example, 'London' is spelled 'Lontoo' in the nominative case, but there is an abundant number of variations, such as 'Lontoossa' ('in London'), 'Lontoon' ('London's'), 'Lontooseen' ('to London'), 'Lontoosta' ('from London'), 'Lontoolaisen' ('Londoner' / 'of London'), etc. Finnish has about fifteen cases[8].

Due to our practical restrictions and our philosophy of using mainly language-independent methods, we cannot build morphological analysis tools for each of the languages involved. Instead, we decided to produce simple stemmers consisting of regular expressions listing all possible place name suffixes and, for agglutinative languages, suffix combinations. A place name such as 'Lontoolaisen' will thus be recognised because the word matches the combination of 'Lontoo' and one of its possible suffixes 'laisen'.

We are planning to use this shallow stemmer also to recognise place names by the names of their inhabitants (e.g. for the Canadian province 'Alberta', the inhabitants and adjectives are 'Albertan / Albertans' in English and 'Albertain / Albertains / Albertaine / Albertaines' in French). For most towns, in French, there is a noun to refer to its people and, according to [11], 95% of the city inhabitant names can be produced using 10 rules. Using such language-specific rules is important if the aim is to achieve a good recall, because we cannot expect our database to contain all inhabitant names.

For most languages we have been dealing with so far, place names must be written in upper case. Therefore, we only check for upper case words whether they are potential place names. This makes the recognition process faster and avoids many wrong hits because many place names are also frequent words in

[8] For an overview of Finnish declensions: http://www.cs.tut.fi/~jkorpela/finnish-cases.html

that language (e.g. French noun 'tours' – 'turns' in English - vs. the city 'Tours'). However, not all languages use upper case letters or write place names in upper case (e.g. Arabic, Hindi, Chinese, Japanese), furthermore, applying our tool on the result of a speech recognition means the absence of uppercases. Therefore, our PERL tool provides the parametrisable and language-specific option whether to restrict the search to upper case words or not.

As many place names are multi-word expressions (e.g. 'San Diego', 'San Marino', 'New York'), our tool parses the text and checks for each upper case word whether it is either a place name by itself, or whether it is the beginning of a multi-word place name. In the latter case, a regular expression checks whether the right-hand-side context of the first word found (e.g. 'San') contains the remainder of the multi-word name (e.g. 'Marino', 'Diego', etc.).

4.2 Geo-Coding

After the recognition of potential place names during the geo-parsing step, these potential place names need to be disambiguated and linked to the relevant data base information.

4.2.1 Disambiguation of Potential Place Names

As shown in section 2, resolving ambiguities is not a trivial task. The main difficulties are linked to:

(a) Place names that are also words in one or more languages, such as 'And' (Iran) and 'Split' (Croatia);

(b) Place names that are homonymic with people's names, such as 'Victoria' (Hong Kong and others) and 'Annan' (UK);

(c) Places that have varying names in different or even in the same language ('Saint Petersburg', 'Saint Pétersbourg', 'Санкт-Петербург' [Sankt-Peterburg], 'Leningrad', 'Petrograd', etc.).

(d) Multiple places that share the same name, such as the fourteen cities and villages in the world called 'Paris';

Our tool relies on a simple dictionary lookup in the text, which means that issue (a) can only be solved using extensive language-dependent geo-stop-lists (namely a list of location names that our system should never recognise. Examples like the frequent words 'And' and 'Split' make it clear that geo-stop-lists are language-dependent because the same ambiguities do not exist in all languages. Such lists do not necessarily have to be generated manually. Instead, they can be created by first extracting all potential place names from a corpus and by then looking at those place names that appear more often than expected. For example, some small village appearing more often than a city like 'London' has very high probability to be a false hit.

The ambiguity between place names and frequent other words of a language only exists when these other words are written with initial upper case letters. A solution is thus to use a corpus to produce frequency lists of pairs of words that occur both in upper case and in lower case (e.g. English 'Split' and 'split'). Words that are homographs with place names that occur much more frequently in lower case than in upper case can thus safely be added to the geo-stop-list as they will raise more wrong hits than good ones. In the future, we are planning to apply more sophisticated rules that make use of the context in order to recognise place names in strings such as 'The City of Split', 'Split, Croatia', 'Split (Croatia)' etc.

A problem with our language-independent approach is the ambiguity between place names and person or organisation names, described above as problem (b). It is possible to put the most frequent ambiguous person names (e.g. 'Bush', 'Chirac', 'Annan') into our geo-stop-list, but common first names like 'Victoria', which can refer to important geographical places, should not be discarded automatically as this would lead us to miss some major locations. The best way to avoid this problem is to previously recognise person names.

Problem (c), caused by the fact that place names have many different translations, can only be solved by completing the database of place names. We have recently merged our database with the Estonian KNAB database and hope to incorporate more sources in the future.

The first attempt to solve problem (d), i.e. the ambiguity between places that share the same name, is to reduce the size of the database. As we were mainly interested in European place names, we reduced the half million place names in our database to about 85,000 by taking out the smaller places outside Europe. The potentially lower recall is acceptable for us because the majority of our text sources is from Europe and the US and tends to mention the country name when talking about smaller places outside Europe so that the reference to the country is not lost. This reduction improved the computational efficiency and reduced noise.

As a result, we then have a number of clearly unambiguous place names such as 'Vladivostok', but we also have some ambiguous ones, such as 'Victoria', which could belong to Canada, to the Seychelles, to Hong-Kong or to over one hundred other places around the world. We try to further disambiguate and to identify one single location for each of these ambiguous place names, by looking at the combination of two parameters:

- The relative 'importance' of the place, according to the size information in our database (the name 'Paris' will preferably be associated to the capital of France);
- The other place names mentioned in the same text: an ambiguous place name will be identified as belonging to a certain country if this country is already being referred to in other parts of the text. In the short text 'Victoria is the business and cultural centre of the Seychelles', 'Victoria' will be geo-coded as the capital of the Seychelles.

Once we have disambiguated a name, we will assume that it will always refer to the same location, relying on Gale et al.'s assumption [7] that there is only one reference per discourse. This means that a place name will have the same reference throughout the text.

Two heuristics will soon be added:

- We will offer the possibility to automatically set a context when launching the geo-coding tool. For instance, an article published in a Hong Kong newspaper will automatically contain Hong Kong in its context, so that every mention of 'Victoria' will, by default, be coded as the Hong Kong capital;

- We will take into account the kilometric distance between a potential place and others, using the co-ordinates from our database, to increase its weights. This could help us, for instance, when a text talks about several places in Portugal and a further place is ambiguous between a Portuguese and a Brazilian town. In this case, it is more likely that the place name refers to Portugal.

4.2.2 *Geo-Coding a Text*

There are various ways to output the extracted information: the basic output is a file in XML format where the place names are marked up with the information available (latitude/longitude, country, etc.). It is then possible to produce a text with the locations highlighted, as shown in Figure 1. We also have the possibility to store this information in a relational database. This last option allows us to translate any kind of query into SQL in order to enhance document retrieval functionality. For instance, it is then possible to query for all Polish texts talking about Hungary and which mention a place that is less than 300 km away from Budapest.

The consignment was intercepted last month on a desert road near the port city of <geo country_id="sa" place_id="3316" class="2" lat="21.5435" long="39.173">Jeddah</geo>. Police say the weapons - capable of being used to bring down aircraft - had been smuggled from <geo country_id="ye" place_id="3261" display_name="Republic of Yemen">Yemen </geo>

Грузовик был остановлен на идущей через пустыню дороге возле портового города Джидда[*Jeddah*]. Полиция утверждает, что ракеты были нелегально переправлены из Йеменаа[*Republic of Yemen*].

Figure 1: Automatically identified place names in an English and a Russian text. The XML tagging can be used to highlight places (the English translation of the place names is given in brackets to facilitate cross-lingual information access).

The database used will thus contain a link between text and place name information. We also store the number of references per country for each document. We use this information to compute efficient queries by country, and also as a feature to calculate cross-lingual document similarity.

The geographical co-ordinates of cities can easily be used to provide a link to other texts mentioning this city, even if they are written in another language. They can furthermore be fed to a map server (like MapQuest[9] shown in Figure 2) providing a detailed map (city level), given latitude and longitude. Calling an external server with latitude/longitude instead of the place name avoids the problem of name variants: we can see in Figure 2 that MapQuest uses 'Jiddah' instead of 'Jeddah'.

[9] http://www.mapquest.com/

Figure 2: After place name identification, the MapQuest server or similar tools can be called automatically by using the co-ordinates of a place (here Jeddah).

5 VISUALISATION

5.1 Producing a Map

Even when displaying a single text on a map, we have to make some choices. The main visualisation parameter will decide on the part of the world map we will display in the automatically generated map (the *bounding box*). We normally display all countries mentioned in the document or the text collection, but if the geographical focus is in one smaller area and there are only a few outliers that are very far away, we will not display the whole map. Instead we zoom in on the zone where at least 90% of the geographical references are situated. This means that up to 10% of the geographical references will not be displayed because they are out of the bounding box, but the map will be in a scale that is more readable.

Another parameter specifies whether the city names should be displayed in the map, or whether place names should only be represented by 'dots'. This parameter sets the number of cities that should feature by name on the map. While it is possible to display all identified place names, maps easily get crowded so that we usually prefer to display only the first 10 by name and to represent the remaining cities only as dots. The third parameter chooses whether the country name and the relative number of hits per country should also appear on the map. Various additional parameters control the look-and-feel of the map, such as the choice of colours, the intensity of country colour depending on their importance, and the size of city dots depending on the number of references in the text.

We have tested various ways to display the maps. These are described in the following subsections.

5.1.1 *Drawing a Map from 'Shape Files'*

The first application we built in-house was using *shape files* for all countries of the world. A shape file contains a set of polygon co-ordinates. Displaying a country on a map consists of drawing the polygons described by these co-ordinates. From such co-ordinates, we produce GIF, PNG or JPG files, using the GD PERL module (Figure 3). The shape files themselves are freely available[10]. This task is both tedious and computationally heavy.

[10] http://www.esri.com/ provides the 'Arcexplorer' tool that allows downloading shape files for countries.

The main disadvantages are that the implementation work is tedious and that the generation of maps is slow and computationally heavy when the polygon generation is not optimised. It requires a shape file for each country. The main advantage is its independence from commercial software.

Figure 3: Map of Saudi Arabia (2 hits) and Yemen (1 hit), generated on the basis of *shape files*.

5.1.2 Using a Specific Program/Server

Visualisation of geographical maps is a common task, and commercial software exists providing such functionality. We produce maps using the *Digital Map Archive* (DMA) tool from the JRC's ISFEREA project (accessible at http://dma.jrc.it/), which is based on the commercial application ArcIMS by ESRI[10]. It allows to overlap multiple layers to add information on roads, airports, lakes, population density and more. It also contains satellite images of the world, at multiple resolutions, and allows thus a very realistic view of the area displayed.

The advantage of using this ready-made tool is obvious: it saves a lot of development time, it is very fast at generating maps, and it allows to include and show various types of information. Due to the fast map generation process, we can even provide the users with an interactive zoom and scroll functionality so that they can explore the full functionality offered by the application (see Figure 4).

The major disadvantage of such a commercial tool is the fact that it is not freely available. DMA is rather easy to integrate as it accepts XML files as input, but exploring the encoding of the information required quite some effort.

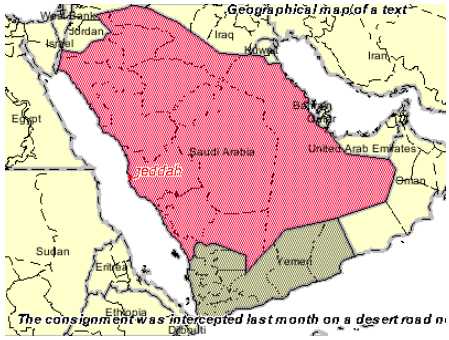
Freely available tools (like *Generic Map Tools*[11], used by [10]) provide limited functionality to display a map given some coordinates.

5.1.3 Using Scalable Vector Graphics (SVG)

We have started to experiment with using the *Scalable Vector Graphics* (SVG) format, which seems to provide a very suitable interface to our maps. The SVG format, defined by the *World Wide Web Consortium* W3C, is a language for describing graphics in XML.[12] On the web, those graphics are displayed using a specific 'plug-in' which then allows the user to zoom and scroll the map. It also allows to produce animations and is generally very suitable for our needs. SVG can be associated with a scripting language to allow some interactivity. Figure 5 shows a screenshot of a SVG map that displays to the user the textual context in which the place name was found when moving the cursor over the map.

SVG files can be generated on the server (server side SVG). The client needs to install the plug-in to see them. The advantage of this method is that the image can be generated quickly (on the fly) by the server. Another clear advantage is that we can provide SVG maps with some interactive animations (showing/hiding cities, showing/hiding some countries, …), and that the user can zoom and scroll the map without overloading the server.

The disadvantage of SVG maps is that zooming in on smaller areas does not create nice graphics. The SVG maps produced for Luxembourg state, for example, are of lower quality than those produced by DMA (see Figure 4).



The consignment was intercepted last month on a desert road near the port city of Jeddah. Police say the weapons - capable of being used to bring down aircraft - had been smuggled from Yemen.

Figure 4: A short text including geographical information and two different visualisations produced with DMA.

Figure 5: Example of an interactive map, using SVG.

[11] http://gmt.soest.hawaii.edu/

[12] http://www.w3.org/Graphics/SVG/

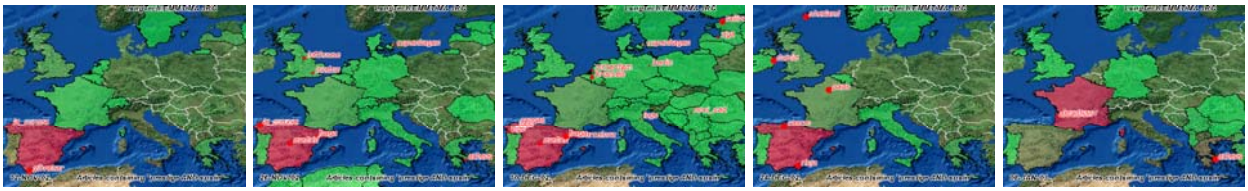


Figure 6: Animation example: "Prestige" tanker accident in Spain (11/2002). The geographical news coverage shows that the oil moved towards the coast of France in January 2003.

5.2 Animations

As we mentioned in section 3, we have started using this tool for the visualisation of newspapers articles. One specific need in news analysis is the tracking of geographical coverage of news on a certain subject over time[1]. In addition to showing the users a sequence of maps, we also produce *animated maps* that are generated by first producing maps for smaller periods (typically days or weeks), and by then concatenating the single images. An example of such animation is shown on Figure 6. Image transition can be smoothed to make the viewing more pleasant and to enhance the feeling of development over time. It goes without saying that the bounding box of the images of an animation has to be the same for all. We use the freeware *ImageMagick* (www.imagemagick.org), which includes tools to create animated GIF or MPeG animations.

Creating animations is a quite a slow process, especially when producing an animation of many images. It requires the time to create each image plus the time to combine them. Using SVG for animations seems to be even more promising, as its language was especially created for animations.

6 EVALUATION

Evaluation of such a system is not trivial as not all of its outcome is easily measurable. According to [15], such an evaluation can be *intrinsic* (i.e. evaluation of the number of correct and incorrect places recognised), or *extrinsic* (i.e. evaluation of the advantage such maps provide to the user).

Regarding the correct recognition of place names in text, we have launched a quantitative evaluation in eight languages: Danish, English, Spanish, Finnish, French, Italian, Swedish, and Russian. The latter has only become useful recently because we have now merged the multilingual spelling variants of place names from the KNAB database with our own. We have chosen to evaluate Russian place name recognition because it uses the Cyrillic alphabet and has a quite complex declension system, which makes it more challenging and interesting. The results are shown in Table 2.

Our manual quantitative evaluation covers 48 parallel texts, (official European Union documents in *en*, *es*, *fr* and *it* languages, some of them also in *da, fi and sv*) and 25 additional texts from newspaper in Russian. The evaluators were asked to judge the correctness of the geographical coding (i.e. the countries and cities found at the text level). This test set represents about 490 KB and contains altogether about 1650 geographical references. It yielded an average precision of 98% and a recall of 88%. These are good results, most of the texts contained the same common place names ('United Kingdom', 'London'…), which makes recognition easier. The texts were similar to newspaper articles, which is the major text type we want to work on. It is possible that for other kinds of text (like tourist information or local newspapers), the results would be less good. The place names not recognised by our software (lowering recall) were mainly due to a problem of declension on multi-word place names (for Russian and Finnish in Table 2) and locations that were not in our database (e.g. the small Caribbean territory 'Montserrat', the Iraqis cities of 'Tikrit' and 'Karbalā'). The wrongly recognised place names (lowering precision) were mainly person or organisation names that have homographs with place names.

We have not launched a complete evaluation on all languages because it is almost impossible to find parallel texts in 20 languages, and our software is not entirely ready to be evaluated for some languages having a strong declension system, because the declension list has to be validated by an expert for each language. Some languages are well covered in our database for their own local cities, but exonyms are often missing. The Greek names in our database, for instance, cover mainly the Greek cities.

The evaluation of the visualisation of place names in maps would be *extrinsic*: we should evaluate what is the impact of such maps and animations on the effectiveness of an expert. However, this is a difficult task that would require a willing user group. However, the users to whom we have presented the functionality of the visualisation were positively impressed. The animations of places mentioned in articles over time raised a particular interest.

7 CONCLUSION AND FUTURE WORK

We have presented our system to recognise and visualise textual geographical information in its current state. Our focus is not on optimising the results for a specific language. Instead, it is im-

Table 2: Evaluation results

	da Danish	**en** English	**es** Spanish	**fi** Finnish	**fr** French	**it** Italian	**sv** Swedish	**ru** Russian	**Total**
#texts	29	48	48	18	48	47	18	25	281
Precision	99%	98%	99%	100%	99%	96%	100%	92%	98%
Recall	86%	90%	93%	77%	94%	86%	80%	93%	88%

portant for us to have a system that works rather well for many languages. The data from the KNAB database helped us to improve the coverage for languages other than English quite a lot.

The visualisation of automatically generated maps, maps of sets of texts, or animated maps is a new field of interest. It is particularly useful for international organisations, like ours, where the language barrier is a day-to-day challenge. New technologies (like SVG) allow us to provide the users with clear and compact information, like an animated world map on the news coverage for a specific event[1].

We are aware that there is a lot of room for improvements. Our future work will consist of:

- Improving the content of the dictionary, by finding new sources or improve the content by learning place names from corpora (querying the Web, using parallel texts, …);
- Studying better ways to identify person names in order to avoid the confusion between place and person names (e.g. "Mr. Orville *London* was elected Chief Secretary of the Tobago House of Assembly");
- Improving the visualisation (animations, interactivity) by using SVG;
- Using the geo-coding information to improve statistical measures for cross-language document similarity, for clustering, classification, topic detection and tracking, cross-language information retrieval, geography-based question answering, etc.

8 ACKNOWLEDGEMENT

We would like to thank Peeter Päll, from the Institute of the Estonian Language in Tallinn, who kindly provided us with the KNAB geographical database.

Thank you also to all the team of the Web Technology sector of the JRC, who had the idea of using SVG graphics for maps, and provided the incomparable data of the *Europe Media Monitor* [1]

We also want to thank Juha Ovaskainen who kindly helped us finding our way through the jungle of Finnish declensions.

9 REFERENCES

[1] Best, C., van der Goot, E., de Paola, M., Garcia, T., and Horby, D. Europe Media Monitor – EMM, Technical Note No I.02.88, Internal document, IPSC, Joint Research Centre, Ispra, Italy, September 2002

[2] Bilhaut, F., Charnois, T., Enjalbert, P., and Mathet, Y. Geographic reference analysis for geographic document querying. Workshop on the Analysis of Geographic References at the NAACL-HLT 2003 conference (Edmonton, Canada, May 2003)

[3] Christel, M.C., Olligschlaeger, A.M., and Huang, C. Interactive Maps for a Digital Video Library, IEEE Multimedia, Vol. 7(1), Jan-Mar 2000, 60-67

[4] Daille, B., and Morin, E. Reconnaissance automatique des noms propres de la langue écrite: les récentes réalisations. In: D. Maurel & F. Guenthner: Traitement automatique des langues Vol. 41/3. Hermes, 2000. 601-623

[5] Densham, I., and Reid, J. A Geo-coding service encompassing a geo-parsing tool and integrated digital gazetteer service. Workshop on the Analysis of Geographic References at the NAACL-HLT 2003 conference (Edmonton, Canada, May 2003)

[6] Friburger, N., and Maurel, D. Textual Similarity Based on Proper Names. (MFIR'2002) at the 25 th ACM SIGIR Conference (Tampere, Finland, 2002) 155-167

[7] Gale, W., Church, K., and Yarowsky, D. One sense per discourse. In Proceedings of the Fourth DARPA Speech and Natural Language Workshop (Pacific Grove, California, February 1991) 233-237

[8] Gey, F. Research to Improve Cross-Language Retrieval – Position Paper for CLEF. In Carol Peters (ed.): Cross-Language Information Retrieval and Evaluation, Workshop of Cross-Language Evaluation Forum, CLEF 2000, Lisbon, Portugal. Lecture Notes in Computer Science 2069, Springer.

[9] Hyland, R., Clifton, C., and Holland, R. GeoNODE: Visualizing News in Geospatial Environments. In Proceedings of the Federal Data Mining Symposium and Exhibition '99, AFCEA (Washington DC, March 1999)

[10] Leidner, J., Sinclair, G., Webber, B. Grounding spatial named entities for information extraction and question answering, Workshop on the Analysis of Geographic References at the NAACL-HLT 2003 conference (Edmonton, Canada, May 2003)

[11] Maurel, D., Piton, O., Eggert, E. Les relations entre noms propres: lieux et habitants dans le projet Prolex, In: D. Maurel & F. Guenthner: Traitement automatique des langues Vol. 41/3, Hermes, 2001 623-641

[12] Mikheev, A., Moens, M., and Gover, C. Named Entity Recognition without Gazetteers, In Proceedings of EACL '99 (Bergen, Norway, June 1999)

[13] Pastra, K., Maynard, D., Hamza, O., Cunningham, H., and Wilks, V. How Feasible is the Reuse of Grammars for Named Entity Recognition? In Proceedings of the 3rd LREC (Las Palmas, Canary Islands, Spain, June 2002)

[14] Piton, O., and Maurel, D. Les Noms Propres Géographiques et le Dictionnaire Prolintex, les lieux situés hors de France. in Proceedings of the 4th INTEX workshop (Bordeaux, France, June 2001)

[15] Sparck-Jones, K., and Galliers, J. Evaluating Natural Language Processing Systems: an Analysis and Review. Lecture Notes in Artificial Intelligence 1083. Springer-Verlag, 1996.

[16] Woodruff, A.G., and Plaunt, C. GIPSY: Georeferenced Information Processing System. Journal of the American Society for Information Science, Vol. 45 No. 9, 1994, 645—655