

# Geo-Coding: Recognition of geographical references in unstructured text, and their visualisation

D i p l o m a   T h e s i s

at the

University of Applied Sciences Hof  
Department of Computer Science and Technology  
Technical Computer Science

Submitted by

Marco Kimler

European Commission  
Joint Research Centre  
Via E. Fermi, 1 - TP 267  
21020 Ispra (VA), Italy

Submitted to

Prof. Dr. Richard Göbel

Ispra, 23rd August 2004

# Abstract

The recognition of geographical references in texts is a well-studied field. However, most work focuses on the detection of geographical references, and does not take into account the mapping of references in a text to their real-world counterparts. The papers which also take into account this geo-coding use rather specific approaches, focus on a few aspects, or are not multi-lingual.

The goal of this work is to create a geo-coding approach which can reliably recognise geographical references, and which can easily be used for many different languages. Therefore, four place name filtering and five disambiguation heuristics are combined to a powerful approach. Since the heuristics restrict the use of linguistic analysis to a minimum, the approach is largely language-independent. It is shown that these enhanced and new heuristics clearly outperform the approach presented in Pouliquen et al. (2004), on which this work is based. The performance of the heuristics is comparable for all five languages analysed. A new two-pass querying approach, called shallow-deep parsing, is introduced, which significantly enhances the results.

Furthermore, this paper presents a dynamic and interactive visualisation of the detected references based on Scalable Vector Graphics (SVG). In addition to basic features like scrolling and zooming, this visualisation offers previously unseen features like the display of the context of a detected reference and the possibility to show alternative places sharing the name with a detected reference.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Aims and objectives . . . . .	2
1.2	Thesis outline . . . . .	3
<b>2</b>	<b>Background</b>	<b>4</b>
2.1	Geo-Parsing - Extracting geographical references from texts . . . . .	4
2.2	Geo-Coding - Mapping references to real-world counterparts . . . . .	6
2.2.1	Place name filtering . . . . .	6
2.2.2	Place name disambiguation . . . . .	8
2.3	Visualisation . . . . .	9
2.3.1	Scalable Vector Graphics (SVG) . . . . .	10
2.3.2	Other vector formats . . . . .	11
2.4	Related work . . . . .	12
2.4.1	Natural Language Processing (NLP) and Geo-Parsing . . . . .	12
2.4.2	Geo-Coding . . . . .	12
2.4.3	Visualisation of geographical references . . . . .	13
<b>3</b>	<b>A heuristical approach to geo-coding</b>	<b>14</b>
3.1	Methodology . . . . .	14
3.1.1	Gazetteer . . . . .	14
3.1.2	Geo-Parsing . . . . .	16
3.1.3	Geo-Coding . . . . .	17
3.1.4	Test sets . . . . .	17
3.1.5	Analysis and evaluation . . . . .	19
3.2	Geo-Coding heuristics . . . . .	21
3.2.1	Context extraction . . . . .	21
	Place of publishing . . . . .	21

Place of writing . . . . .	22
Shallow parsing of the text . . . . .	23
3.2.2 Shallow-deep parsing . . . . .	23
3.2.3 Place name filtering . . . . .	25
Heuristical person name detection . . . . .	25
VIP lists . . . . .	26
Geo-stop-lists . . . . .	27
Place triggering . . . . .	27
3.2.4 Place name disambiguation . . . . .	28
Relative importance of places . . . . .	29
Context-based triggering . . . . .	29
Comparison of a location to other place names in the text . . . . .	30
Distance of locations from the event . . . . .	30
3.2.5 Weighing the heuristics . . . . .	33
3.3 Results . . . . .	35
3.3.1 Performance of the shallow deep parsing . . . . .	36
3.3.2 Performance of the place filtering heuristics . . . . .	37
3.3.3 Performance of the place disambiguation heuristics . . . . .	40
3.3.4 Different languages . . . . .	41
3.3.5 Topics . . . . .	43
3.3.6 Running time analysis . . . . .	46
3.3.7 Summary of the results . . . . .	48
<b>4 Visualisation</b>	<b>50</b>
4.1 Data model for visualisation . . . . .	52
4.1.1 SVG data . . . . .	52
4.1.2 Programming logic . . . . .	53
4.2 Visualised features . . . . .	55
4.2.1 Overview . . . . .	55
4.2.2 Country shapes and background . . . . .	56
4.2.3 Detected place names . . . . .	57
4.2.4 Context of the geographical references . . . . .	59
4.2.5 Place alternatives . . . . .	60
4.2.6 User interface . . . . .	61
<b>5 Discussion</b>	<b>62</b>

<b>6 Conclusion</b>	<b>65</b>
6.1 Contributions . . . . .	66
6.2 Future work . . . . .	67
6.3 Outlook . . . . .	68
<b>A Results in tables</b>	<b>69</b>
<b>Bibliography</b>	<b>75</b>

# Abbreviations

DMA	Digital Map Archive, a GIS developed by the JRC
DOM	Document Object Model
ECMA	European Computer Manufacturer's Association
GIS	Geographical Information System
JRC	Joint Research Centre of the European Commission
NER	Named Entity Recognition
NLP	Natural Language Processing
SVG	Scalable Vector Graphics
UTF-8	Unicode Transformation Format 8
VRML97	Virtual Reality Modelling Language, as specified in ISO/IEC 14772
X3D	Extensible 3D
XML	Extensible Markup Language

# Chapter 1

## Introduction

In the last ten years, the Internet has literally exploded. This applies on the one hand to the number of its users, but also to the amount of information which is available. Consequently, the major problem is not to obtain the desired information, but to identify those parts of it which are relevant and reliable (Wikipedia 2004c). A good example of the abundance of information are news articles: although most of the well-known news agencies and media companies provide articles on the web, it is rather knotty to quickly and reliably find information about a certain subject or region. An automated approach, which would be able to cluster newswire texts by topic or geo-context, could help to keep track of those parts of the information the reader is interested in.

The JRC's language technology group has created a *Top stories* application, which automatically analyses newswire texts and groups them into news clusters of the same topic. A related issue is the grouping of texts with the same geo-context, i.e. texts which are about the same regions or countries. On the underlying detection of geographical references some research has already been done (Li et al. 2003; Leidner, Sinclair and Webber 2003; Pouliquen et al. 2004), but the disambiguation approaches these papers use are rather specific, and we think that an improvement and combination of them could significantly enhance the results. Furthermore, most of the approaches are language-dependent, and therefore are not usable in a multi-lingual environment. This, however, is a prerequisite for many applications at the European Commission. Hence, the goal of this work is to create a system which can reliably detect geographical references, and which can easily be used for many different languages.

Only little work has been done on the visualisation of geographical references. Most papers just use online services, which very well can display locations on a map, but are

not able to provide additional information, like the context a reference occurred in. Interactivity is either not supported at all or provided in a rather rudimentary form. In that aspect, much of the potential of a software-based map is left unused.

This work will focus on both issues: at first, a heuristic-based multi-lingual approach to the recognition of geographical references in texts will be presented, and its performance will be evaluated. Then, the detected geographical references will be visualised in an interactive, easy-to-use map.

## 1.1 Aims and objectives

*The aims of this work are to recognise geographical references in unstructured texts, and to relate these references to actual place names (geo-coding). The focus lies on the improvement of methods to resolve ambiguities in the detected references. Furthermore, the detected geographical references should be visualised in a map.*

The following objectives are steps towards fulfilling the aim stated above:

1. *Detection of potential place names using an existing gazetteer:*

Gazetteers are lists containing information on geographical references (e.g. name, name variations, coordinates, class, size, additional information) (Leidner, Sinclair and Webber 2003) and are often used for place name recognition. Such a gazetteer should be used also for this work. Since the resulting program should be used for a variety of languages, the detection of potential place names must be multi-lingual, and must also support different spellings of one place name (Saint Petersburg = Saint Pétersbourg = Санкт-Петербург = Leningrad).

2. *Extending data:*

The gazetteers used in previous work are rather small and often contain less than 100,000 entries (Ignat et al. 2003; Leidner, Sinclair and Webber 2003; Pouliquen et al. 2004). The gazetteer used at the JRC and described in Pouliquen et al. (2004), for example, contains only 10,000 place names, and (outside Europe) only covers bigger places. This often circumvents cases of ambiguity because many places of the same name do not (or only seldom) occur in the database, but it also results in a lower recall (Pouliquen et al. 2004). However, this basic gazetteer may serve as a baseline for evaluation (see also below). The gazetteer should be extended to the 500,000 place names which are already included in other gazetteers available at the JRC.

3. *Analysis of these potential place names and resolution of ambiguities:*

Step 1 just detects that *Paris* is a location. It is not clear whether it is referred to *Paris*, the capital of France, or *Paris* in Kentucky (USA). Furthermore, *Annan* might be identified as a place in the UK, while the text is about the U.N. Secretary-General, Kofi Annan. Therefore disambiguation methods for resolving such ambiguities are to be defined and implemented. This objective will be the main part of the work.

4. *Evaluation of the defined disambiguation methods on newswire texts:*

The efficiency of the mentioned disambiguation methods is to be evaluated by analysing newswire texts. Baseline is the analysis of the texts with the heuristics presented in Pouliquen et al. (2004). Based on these results, the contribution of each method used in this work has to be evaluated.

5. *Visualisation of the disambiguated place names on a map:*

As the gazetteers also store latitude and longitude values for the locations, it is possible to show the detected references on a map. Such a map-based visualisation should be developed. The visualisation could be interactive, i.e. could allow the user to zoom and scroll the map, and to change the amount of information shown. Possible techniques are Scalable Vector Graphics (SVG) or the use of ready-made tools or services like the Digital Map Archive (DMA).

## 1.2 Thesis outline

The rest of this thesis is structured as follows: Chapter 2 will provide some general background on geo-parsing, geo-coding, and visualisation. Furthermore, related work done in these fields is presented. Chapter 3 focuses on the heuristical geo-coding approach presented in this thesis: it outlines the general methodology used (e.g. information about the gazetteer, the test sets, and the evaluation methodology), describes the single heuristics utilised in this work, and presents the results from applying these heuristics to 161 newswire texts. A new map-based visualisation for geographical references will be introduced in chapter 4. The results presented for the geo-coding approach and the visualisation are discussed in chapter 5. Chapter 6 presents the most important conclusions drawn from the results, together with a summary of the contributions of this thesis and ideas for future work. Finally, an outlook will be given of how the presented approach will be used at the JRC.

## Chapter 2

# Background

The process of recognising geographical references in texts can be divided into two phases: at first, the text to be analysed is parsed, and words which are geographical references are extracted. This phase is called geo-parsing and will be further introduced in section 2.1. The second phase - referred to as geo-coding - includes the mapping of references (e.g. the word *Paris*, which is found to be a place) to their real-world counterparts (the place which is known as the capital of France). The geo-coding involves a filtering of words which are no geographical references in this context, and a disambiguation between locations sharing the same name. See section 2.2 for more information on geo-coding. Section 2.3 will give some background on the visualisation of geographical references. Finally, an overview of related work on geo-parsing, geo-coding and visualisation will be presented in section 2.4.

### 2.1 Geo-Parsing - Extracting geographical references from texts

Geo-parsing refers to the extraction of place names in texts (Pouliquen et al. 2004). Geo-parsing is related to the field of Named Entity Recognition (NER), which in general deals with the detection of named entities. For geo-parsing, the NER-approaches focus at proper names in general, and locations in particular.<sup>1</sup>

---

<sup>1</sup>Chinchor (1997) define three types of names entities: temporal expressions (e.g. dates and times), quantities (monetary values and percentages), and proper names. Proper names are subclassified into organisations, persons' names, and locations (Chinchor 1997).

Basically, there are two kinds of NER systems (Tjong Kim Sang and De Meulder 2003):

- *Internal approaches*

Internal approaches analyse a text by trying to extract named entities without referring to other, external resources. The most frequently used internal approaches are techniques like artificial neural networks (ANNs), hidden Markov models (HMMs), and Maximum Entropy Models. These approaches try to automatically extract features by being exposed to a set of training data. Other internal techniques use more supervised approaches like grammars, or part-of-speech tagging (Tjong Kim Sang and De Meulder 2003).

After being trained or being optimised for a certain language or type of texts, internal approaches can detect even previously unseen data. However, the performance depends largely on the size and quality of the training set, and an extensive individual adaptation and training is necessary to support new languages.

- *External approaches - Gazetteers*

Here an external resource, a so-called *gazetteer*, is used. Gazetteers are lists containing information on geographical references (e.g. name, name variations, class, size, coordinates), which are lately usually stored in databases (Leidner, Sinclair and Webber 2003). If such a gazetteer is queried, all words in a text, which are also places somewhere in the world, are returned.

The advantage of gazetteers is that they are generally language-independent, provided that they contain also language-specific spellings and alternative character sets. However, only references stored in the gazetteer can be detected, and not all hits may really refer to a place in the particular context (see also section 2.2.1 for more details on that).

Generally, internal approaches may very well detect that a word in a text is a place name, but they cannot provide additional information, like size of the place or its coordinates. Since place names - in opposition to person names - rarely change and new place names seldom occur, a rather static database (the gazetteer) can be used to get this information. Since Mikheev, Moens and Grover (1999) report that gazetteer-based techniques lead to significantly higher precision and recall values than purely internal approaches, the usage of a gazetteer for place name recognition seems clearly favourable.

## 2.2 Geo-Coding - Mapping references to real-world counterparts

Once detected, the found references have to be mapped to the real-world counterpart they refer to. If in the text the word *Paris* is found to be a geographical reference, it has to be assigned to the “real” place *Paris*, which is the capital of France (or to one of the other Paris in the world - see section 2.2.2 for more details). Leidner, Sinclair and Webber (2003) call that mapping of words and texts to real-world places *spatial grounding*. Most other papers refer to this mapping, together with a further annotation of the found reference (e.g. with latitude/longitude data, place class etc.) as *geo-coding* (Densham and Reid 2003; Pouliquen et al. 2004).

Since many entries in the gazetteer are ambiguous, the mapping of a potential geographical reference to its real-world counterpart is nontrivial. Simply speaking, there are two types of disambiguation problems to be dealt with:

- As indicated in section 2.1, not all detected references are really places in the real world. There are many place names which are also words in one or more languages, or which are homonymic with persons’ names (Pouliquen et al. 2004). See section 2.2.1 for more details on this place name *filtering* issue.
- Many place names are not unique. Especially many European metropolises have counterparts in the “New World”, where immigrants gave their newly built up settlements names of their home countries. This issue of *disambiguating* place alternatives will be further expanded in section 2.2.2.

### 2.2.1 Place name filtering

Many place names are ambiguous in the sense that they are homonymic with other words in natural language, or with persons’ names. That is mostly due to the fact that people gave their settlements “telling names”. One well known (but certainly uncommon) example is *Llanfairpwllgwyngyllgogerychwyrndrobwlllantysiliogogoch* in Wales, which is one of the world’s longest place names and translates as “St Mary’s church in the hollow of the white hazel near a rapid whirlpool and the church of St Tysilio of the red cave” (Fatman and Stewart-Noble 2000). But also more important places like *Yerusalem*, Israel (“City of peace”) or *Vladivostok*, Russia (“Rule the East!”) are telling names.

While these places were intentionally given a describing name, others share their name with words in some language by coincidence. Especially many short and frequent words

are affected: As table 2.1 shows, at least 10 of the most frequent 30 words in English, German and French are also place names somewhere in the world. Similar problems arise virtually any type of word, e.g. colours (*Black*, Montana, United States, *Rouge*, France, *Blau*, Namibia), date specifications (*Friday*, Texas, United States, *May*, Niger, *Winter*, Canada) etc. This problem is especially apparent for geo-parsing approaches which do not apply Named Entity Recognition (NER) techniques (see also section 2.1).

Another problem are place names which are homonymic with names of persons or organisations. Table 2.2 shows some examples with ambiguous VIPs' names from politics, sports, and entertainment. Also in this case, NER techniques could detect persons' names, but the problem remains for cases where NER cannot be utilised, for example in a multi-lingual environment.

English		French		German	
And	Ireland	De	Burkina Faso	Die	France
To	Ghana	Du	Ghana	Den	Ethiopia
Be	India	Un	Russia	Zu	Zaire
By	Sweden	Une	Colombia	Ist	Hungary
Are	Nigeria	Est	Netherlands	Im	Russia
This	France	Il	Iran	Dem	Cameroon
But	Afghanistan	Au	Austria	Als	Denmark
Had	Oman	Par	Great Britain	Auch	France
She	India	Sur	Oman	An	Mexico
We	Zaire	Pas	Turkey	Aus	Namibia

Table 2.1: At least 10 out of the most frequent 30 words in English, French and German are also places somewhere in the world.<sup>2</sup> An example: *And* is not only a common English word, but also a place in Ireland.

<sup>2</sup>Sources of the frequency lists (All URLs in this paper last checked on 2004-08-23):

English: [http://javelina.cet.middlebury.edu/lisa/top\\_words.htm](http://javelina.cet.middlebury.edu/lisa/top_words.htm)

German: <http://german.about.com/library/blwfreq01.htm?terms=frequency>

French: <http://www.unine.ch/info/clef/wordStatfr.txt>

George Bush	South Africa United States	Lance Armstrong	Mozambique Argentina
Tony Blair	United States Malawi	James Joyce	Zambia United States
Jacques Chirac	– France	Michael Jackson	– United States
Kofi Annan	Mali Great Britain	Jack Nicholson	South Africa Australia
Javier Solana	Spain Philippines	Henry Ford	Haiti Ireland

Table 2.2: Well-known persons who share their names with places all over the world.

Example: *George* is a location in South Africa, *Bush* a place in the US.

### 2.2.2 Place name disambiguation

A common problem of Named Entity Recognition (NER) approaches is that they can detect words which are place names, but they cannot tell which of the place alternatives is referred to. An example: NER-based approaches can detect that *Paris* is a place name, but they do not know whether it is referred to the capital of France, or to one of the more than a dozen Paris in the US, Canada and Gambia.

A look into the Global Discovery Gazetteer, which will also be used in this work (compare section 3.1.1), shows that place names with at least two occurrences are not just a theoretical problem: The gazetteer includes more than 500 places with at least 20 alternatives each, and for many of the world’s major capitals there are more than a dozen places having the same name (see table 2.3). The place name with most occurrences in the Global Discovery Gazetteer is *Aleksandrovka* with 244 occurrences, mostly in Russia and former Soviet Republics.

Place name	N
Aleksandrovka	244
San Antonio	205
Santa Rosa	199
...	
San Francisco	102
Buenos Aires	88
Washington	32
London	18
Berlin	15
Paris	15
Rome	15
Moscow	12

Table 2.3: Some frequent place names and their number of occurrences  $N$  in the Global Discovery Gazetteer.

While pure Named Entity Recognition can get by the use of gazetteers, place name disambiguation definitely needs a gazetteer: Without it, essential additional information like geographical coordinates, mapping to a country or administrative unit, or the importance of the place would be missing.

## 2.3 Visualisation

Generally, there are two systems for representing graphical information on computers: raster graphics and vector graphics (Eisenberg 2002). Raster graphics store the image information in an array of atomic elements, called pixels. If one zooms into such a raster image, the single pixels become clearly apparent, as can be seen in figure 2.1. Vector formats, however, store objects as geometric shapes (lines, rectangles, ellipses etc.) rather than a set of individual pixels. Hence, vector graphics are zoomable without quality loss.

Since this zooming behaviour is indispensable for high-quality interactive maps, the following presentation will restrict to vector formats. Focus will be laid on Scalable Vector Graphics (SVG) as a typical interactive vector format; furthermore, alternative vector formats will be presented. Vector formats which do not support interactions (e.g. PostScript or PDF) and formats which require rather low-level programming (like OpenGL, DirectX,

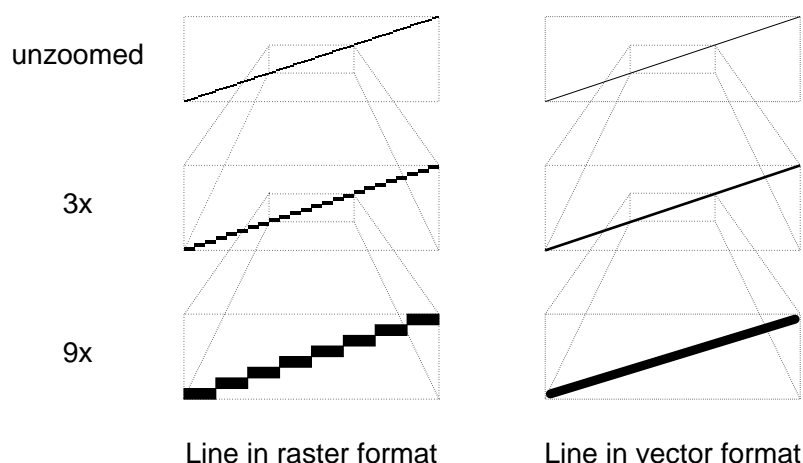


Figure 2.1: Effects of zooming raster and vector objects. If the raster line (left) is zoomed in, the single pixels (which are hardly noticeable in the unzoomed representation) become clearly apparent. The vector line on the right, however, is completely unaffected by zooming.

and Java 3D)<sup>3</sup> will not be described.

### 2.3.1 Scalable Vector Graphics (SVG)

Scalable Vector Graphics (SVG) (W3C SVG Working Group (Ed.) 2003) is a standard for describing two-dimensional graphics in a XML grammar using vector graphics. The following properties make SVG especially appealing for the visualisation of geographical references:

- Since SVG is a XML grammar, the objects are stored in plain text. Hence, SVG images can be easily generated and manipulated, and are human-readable (see figure 2.2). Since XML and SVG are open and widely-used standards,<sup>4</sup> a variety of programs exists for generating, processing, and displaying XML- and SVG-based data.
- SVG implementations include an innate scrolling and zooming behaviour and support animations. As long as this default interactivity is sufficient, the developer does not have to implement any behaviour at all.
- In case the built-in scrolling and zooming functions are not sufficient, the functionality can be extended by ECMA scripts.<sup>5</sup> SVG supports the XML DOM (Document Object Model), which provides a standardised way to access elements, attributes, and properties.
- SVG supports multiple layers by grouping together a number of elements. These groups respectively layers can easily be modified as a whole.
- For viewing SVG graphics, no specialised commercial program is needed. The user only has to download and install a free SVG plug-in, which integrates into the most common web browsers. Currently, the most widely used SVG viewer plug-in is the Adobe SVG viewer.<sup>6</sup>

---

<sup>3</sup>In general, techniques at low-level are more flexible than high-level scripting solutions. However, they are more complex to develop and maintain, and only a fraction of their power is needed. Therefore simpler scripting languages seem advantageous in our context.

<sup>4</sup>Both XML and SVG were developed by the World Wide Web Consortium (W3C), the driving force behind many widely-used open web standards.

<sup>5</sup> ECMA, the European Computer Manufacturer's Association, defined a simple scripting language, which is now commonly known as JavaScript (Eisenberg 2002).

<sup>6</sup><http://www.adobe.com/svg/viewer/install/>

```
<svg viewBox="-8 -59 12 10">
<path fill="lightgrey"
      d=" M0.056,-53.48 [...]" />
<circle cx="-0.18" cy="-51.49"
        r="0.3" fill="darkred" />
<text x="-0.18" y="-51.49"
      font-size="1">London</text>
</svg>
```



Figure 2.2: SVG source code referencing Great Britain and London ( $51.49^{\circ}\text{N}$ ,  $0.18^{\circ}\text{W}$ ), and its visualisation. Note that, due to the SVG coordinate system, latitude values on the northern hemisphere are represented by negative  $y$  values.

### 2.3.2 Other vector formats

Macromedia Flash<sup>7</sup> is a vector format whose features are very similar to those of SVG: it supports interactivity, additional programming logic by using scripts, and only needs a browser plug-in to work. Compared to SVG, Flash has advanced multimedia capabilities (like sound and video) and is more widely used. On the other hand, Flash is a proprietary format and stores its data in binary form, so further processing is more complicated compared to the plain-text format SVG.

VRML97 (Web3D Consortium (Ed.) 2002) and its successor X3D (Web3D Consortium (Ed.) 2003) are both open standards for 3D imaging and lay a focus on virtual reality. Both VRML97 and X3D are powerful standards which do not only support animation and real-time interaction, but also come with advanced features like lighting and texturing (Scheurich 2001). Both formats store information in a text format and run in web browsers after installing a plug-in. However, the 3D formats have high hardware requirements (Scheurich 2001).

---

<sup>7</sup>See <http://www.macromedia.com/software/flash/> for more information on the Macromedia Flash products.

## 2.4 Related work

### 2.4.1 Natural Language Processing (NLP) and Geo-Parsing

Much work has been done on the recognition of geographical references in texts. However, most of these approaches use internal entity recognition techniques and do not allow for subsequent processing phases (compare section 2.1). An overview of recent work on this field can be found in Tjong Kim Sang and De Meulder (2003).

More interesting for this work are gazetteer-based geo-parsing approaches, which prepare the ground for the subsequent geo-coding phase. Mikheev, Moens and Grover (1999) have shown that the use of gazetteers significantly improves the detection results. They conclude that “relatively small gazetteers are sufficient to give good Precision and Recall” (Mikheev, Moens and Grover 1999, p. 8). However, this only holds if the analysed texts mainly contain frequently occurring place names.

If texts also contain seldom place names - which is expected to be the case in this work, since newswire from all over the world is analysed - a big gazetteer is the only chance to detect them. Hence, many other works use larger gazetteers: Leidner, Sinclair and Webber (2003) have used the UN-LOCODE gazetteer provided by the UN with 36,000 entries for their work. Pouliquen et al. (2004) use the KNAB database and the Global Discovery Gazetteer as a basis for their evaluation, but reduced the database size to about 85,000 entries to restrict ambiguity.

### 2.4.2 Geo-Coding

In opposition to NER, little work seems to be done on the field of place name disambiguation. Most of the papers focus on a single or only a few indicators for choosing a place alternative: Li et al. (2003) use maximum weight spanning trees and therefore try to find an alternative which is nearby other detected references; Leidner, Sinclair and Webber (2003) also try to minimise the distances by utilising minimal bounding boxes. Ignat et al. (2003) and Pouliquen et al. (2004) take into account the relative importance of places and the country they are situated in, and are thus focusing on the geo-context of places.

None of the papers examined, however, combines these spatial and more contextual indicators, although Pouliquen et al. (2004) surmise that this might improve the results.

### 2.4.3 Visualisation of geographical references

The visualisation of geographical references is often performed by Geographical Information Systems (GIS), which, in general, are “designed for the acquisition, maintenance, and use of cartographic data” (Tomlin 1990, p. xi). Many GIS, like the ArcGIS by ESRI,<sup>8</sup> provide their own user interface and are not usable for web-based applications. Others, like the *Digital Map Archive* (DMA)<sup>9</sup> (Ehrlich et al. 2003), are able to produce maps in common pixel formats (GIF, JPG, PNG), and have already been used in previous geocoding work (Ignat et al. 2003; Pouliquen et al. 2004). Other, freely available web services like MapQuest<sup>10</sup> can also be used for displaying places with given coordinates in a dynamically generated graphic. However, the generated graphics themselves are not interactive, i.e. it is not possible to zoom or scroll the generated images. In the case of the DMA, this interactivity is provided by generating a new image each time the user requires it, which leads to increased network traffic and hence long response times when using slow Internet connections. Pouliquen et al. (2004) present a visualisation prototype for geographical references using SVG.

Other approaches allow for interactivity. The online routing planner Map24<sup>11</sup> uses Java applets to generate dynamic maps, mappy.com<sup>12</sup> utilises the Macromedia Flash technology. Though these services are free to use, the core is commercial and cannot be used for producing tailored maps with additional information, as aimed at in this paper.

---

<sup>8</sup><http://www.esri.com>

<sup>9</sup><http://www.dma.jrc.it>

<sup>10</sup><http://www.mapquest.com>

<sup>11</sup><http://www.map24.com>

<sup>12</sup><http://www.mappy.com>

## Chapter 3

# A heuristical approach to geo-coding

The aim of this thesis is to recognise geographical references in unstructured text, and to relate these references to their real-world counterparts. This step is called geo-coding. This chapter proposes a set of place name filtering and disambiguation heuristics, which are believed to fulfil the aim, and tests the performance of the heuristics.

The following section will define a methodology, according to which the new heuristics will be applied to test texts; furthermore, the analysis and the evaluation method will be presented. Section 3.2 introduces the heuristics used in this work. Section 3.3 focuses on the results of the application of the heuristics to 161 test texts in five languages.

### 3.1 Methodology

#### 3.1.1 Gazetteer

For this work, a gazetteer which fulfils three basic prerequisites is needed:

- *Support for multiple languages and spellings*

Place names often have different names or spellings in various languages, especially if they are transliterated from non-Latin character sets. Therefore, the gazetteer must contain place names in different languages, spellings and character sets, at least for the most important places.

- *Size of the database*

Since this work analyses newswire texts from all over the world, the gazetteer should

contain as much entries as possible - otherwise less frequent places cannot be detected. Furthermore, a big database also rises ambiguity, and therefore is a bigger challenge for the disambiguation algorithms.

- *Quality of additional information*

Besides the place names, additional information is needed: latitude and longitude values, country and administrative unit the place lies in, information about the importance. Apparently, this data must be of good quality to lead to reliable results.

Two gazetteers were found of which each fulfils at least two of the three prerequisites stated above:

- The KNAB database from the Institute of Estonian Language.<sup>1</sup> Currently, KNAB includes around 83,000 place names with a strong focus on Estonia (33,000 entries). The big advantage of KNAB is its wealth of alternative spellings. Especially for the world's biggest cities it provides a multitude of alternative names in different languages (e.g. *Moscow* (en), *Moskau* (de), *Moscou* (fr), *Moscú* (es), *Mosca* (it)) and various character sets (Москва (Cyrillic), Μόσχα (Greek)).<sup>2</sup> The quality of the additional information seems to be reliable.
- The Global Discovery Gazetteer, compiled by Europa technologies, contains 540,000 unique place names (June 2003). It only contains a few alternative spellings, which are only in Latin character set. However, the entries are distributed all over the world, and much additional information in apparently good quality is provided.

A third gazetteer, the GEOnet names server (GNS),<sup>3</sup> was also taken into account, especially because of its size - it contains about 5.5 million entries. However, an analysis of the data revealed that much of the additional information (especially about importance of places) is missing or incorrect. Therefore, GNS was not used for this work.

To combine the advantages of KNAB (many alternative spellings and character sets) and Global Discovery (many entries), the two gazetteers were merged. Therefore an already existing subset of KNAB, containing around 10,600 entries, was taken as an initial version of the database. Then, all entries from Global Discovery, which were not yet

---

<sup>1</sup>More information on the KNAB is available online from <http://www.eki.ee/knab/knab.htm>

<sup>2</sup>KNAB even includes spellings in Arabic, Japanese, and Chinese. However, these spellings are transliterated versions (e.g. Arabic version of Moscow: *Mūsūkū*). Since this work focuses on European languages which mainly have a Latin character set, this limitation is acceptable.

<sup>3</sup><http://earth-info.nga.mil/gns/html/>

included in the database, were added. The check whether an entry already exists in the database was done by comparing latitude and longitude values as well as the place names to each other. The merging step resulted in a database consisting of 541,000 unique place names with (in total) 601,000 spellings.

The gazetteer is stored in an Oracle 8i database, as character set of the entries UTF-8, a variable-length Unicode encoding, has been chosen. In contrast to the more common ISO-encodings (which focus on a few languages), UTF-8 supports characters from virtually every language - an indispensable feature for a multi-lingual geo-coding approach. Oracle was chosen because it has a stable UTF-8 support, which for example MySQL does not (yet) offer. Since no database server was available, the gazetteer was installed on a Windows XP desktop PC.

### 3.1.2 Geo-Parsing

For detecting potential place names in a text, the geo-parsing approach described in Pouliquen et al. (2004) was used. By default it queries the place name database for each upper-case word in a text. It returns place names if they are written either in the text's language or the local language. That means, in an English text both the English spelling *Munich* and the local German spelling *München* would be detected, but no other variants like *Monaco* (Italian) or *Mnichov* (Czech).

The chosen geo-parsing module is also capable of detecting multi-word expressions, like *New York* or *Jerez de la Frontera*. By using a rule-based heuristic it can also map inhabitant names and adjectives (e.g. *Albertan* (en), *Albertaine* (fr)) to the corresponding place (here: *Alberta*). Furthermore, declensions and suffixes are detected and removed. In Finnish, for example, declensions likes *Lontoossa* (in London), *Lontooseen* (to London), or *Lontoosta* (from London) are all mapped to the nominative case, *Lontoo*, which is the Finnish spelling of *London*. See Pouliquen et al. (2004) for more details on the employed geo-parsing module.

For this work, the geo-parsing module described above was enhanced to support a two-pass query as described in 3.2.2. The result of the two-pass geo-parsing step is an unfiltered list of potential place names. This list contains all words in the text which are either important places (detected in pass one), or are places that are in the geo-context of the text (detected in pass two) - see section 3.2.2 for details. However, the list still may contain false hits, i.e. words which are in general place names, but not in the context they occurred in the text. These false hits are to be sorted out in the subsequent geo-coding phase.

### 3.1.3 Geo-Coding

The geo-parsing just provides a list with *potential* place names. It is not known yet if a potential place is really a place in the context, or to which real place a reference points. These issues are tackled in the subsequent geo-coding step.

The geo-coding approach used resembles the one presented in Pouliquen et al. (2004) and uses heuristics to filter and disambiguate geographical references. In comparison to Pouliquen et al. (2004), the heuristics were enhanced and new heuristics were added - section 3.2 presents the used heuristics in detail.

The heuristics were all implemented in object-oriented Perl. Perl has the advantage that it can be used in both a batch environment (for e.g. processing all incoming texts during a day) and in a web-environment (for processing a text the user clicked on). Since version 5.8 Perl supports UTF-8 and therefore can also handle texts in non-Latin character sets (as Cyrillic and Greek, but also Arabic, Chinese and Japanese). The object-oriented features of Perl were used to allow not only for an easy and reliable development, but also for later reuse, flexibility, and extensibility.

### 3.1.4 Test sets

Since no properly annotated parallel test set for multiple languages could be found,<sup>4</sup> newswire texts in different languages were chosen and manually annotated (see section 3.1.5).

#### Choosing test topics

As test sets, newswire texts on four topics in five different languages were chosen. The texts of each topic were clustered by the *Top stories* application described in Pouliquen, Steinberger and Ignat (2004), which analyses and groups around 450 news sources in five languages (English, German, French, Italian, and Spanish). The selected topics are:

- Arrest of Islamic militants believed to have been planning a bomb attack during a NATO summit in Istanbul (cluster's main article from *Financial Times*, 3rd May 2004).

---

<sup>4</sup>Although there are corpora with annotated place names (e.g. the Reuters corpus<sup>5</sup> or the ECI Multilingual Text Corpus<sup>6</sup>), none of them includes a mapping of places in a text to their real-world counterparts. Hence, with these corpora it would only be possible to test the geo-parsing, but not the geo-coding phase.

<sup>5</sup><http://www.reuters.com/researchandstandards/corpus/>

<sup>6</sup><http://www ldc.upenn.edu/>

- Chechen President Kadyrov killed in a bomb explosion in Grozny (*EuroNews*, 9th May 2004).
- Earthquake in Northern Iran (*Guardian*, 29th May 2004).
- Former US-president Ronald Reagan died (*Tehran Globe*, 6th June 2004).

The *Turkey* and *Chechnya* clusters were chosen as development sets, i.e. these texts were already used while developing the heuristics, and were taken as a basis for fine-tuning the parameters. Since it cannot be excluded that the parameters were unintentionally optimised to fit these texts, the results for these texts must be taken with care.

The remaining two clusters (*Iran* and *Reagan*) were used as test sets. That means, they were chosen after the fine-tuning phase of the parameters was already completed. Consequently, the results cannot have been optimised for these texts, and so the results for these test sets can be estimated more significant.

### Choosing test texts for each topic

Besides the cluster's main article, up to 9 other articles per language were chosen.<sup>7</sup> If there was more than one text from a single news source, only the one appearing first in the cluster was selected. That means, per cluster and per language a maximum of 10 articles coming from different news sources were selected. See table 3.1 for an overview of the clusters and the number of (selected) articles they contain.

Topic/cluster	Number of (selected) texts in					
	German	English	Spanish	French	Italian	Total
Turkey	67 (10)	20 (10)	11 (7)	11 (5)	3 (3)	110 (35)
Chechnya	110 (10)	63 (10)	35 (10)	36 (10)	61 (10)	305 (50)
Iran	32 (10)	13 (10)	6 (5)	9 (4)	9 (5)	69 (34)
Reagan	64 (10)	121 (10)	23 (8)	15 (6)	18 (8)	241 (42)

Table 3.1: Number of clustered articles per language in four clusters of the *Top Stories* application. The number of articles chosen for this work is given in parentheses.

<sup>7</sup>If a cluster contained less than 10 articles in a language, one article from each news source was chosen.

### 3.1.5 Analysis and evaluation

In order to evaluate the performance of the heuristics, each of the 161 development and test texts was manually annotated. Then, the texts were analysed with the heuristics presented in this work. By comparing the outcome of the analysis with the annotation, precision, recall and F-Score were calculated as performance measures.

**Manual annotation of the test texts.** In order to be able to evaluate the performance of the disambiguation heuristics, correctly annotated representations of the test texts had to be created.

In a first step, all texts were automatically analysed by the heuristics described in Poulliquen et al. (2004). The detected place IDs and the number of their occurrences were stored in a hash. This led to a first, but still erroneous representation of the geographical references. Therefore, in a second step the annotations of all 161 test texts were manually checked and corrected: False positives were sorted out, undetected places were added to the annotation, and incorrect assignments were corrected. Three different types of geographical references are differentiated in the annotation: undeclined upper case words (like *France*), declined upper case words (like *French*), and declined lower case words (like the German word *französisch*). In this study, only undeclined and declined upper case words were taken into account, undeclined lower case words were ignored.

The resulting list of place IDs and the number of their occurrences (grouped by the three reference types) were stored in the annotation hash, which is now a correct representation of the text.

**Analysis.** After the geo-parsing phase, the place filtering and disambiguation heuristics are applied to the text. For choosing an alternative, the ‘one sense per discourse principle’ (Gale, Church and Yarowsky 1992) is applied, which states that “if a polysemous word [...] appears two or more times in a well-written discourse, it is extremely likely that they will share the same sense” (Gale, Church and Yarowsky 1992, p. 233). That means, that all occurrences of a place name mentioned in a text refer to the same location throughout the whole text.

The results of each heuristic are combined and weighed according to the weight functions given in section 3.2.5 on page 33. The result of the analysis is a geo-coded text in which each detected geographic reference is assigned to a real-world counterpart. The detected place names are now compared to the manually annotated references. This leads to a quantification of correctly detected references, false hits, and missed locations, which are the basis for calculating precision, recall, and F-Score.

**Comparison of analysed test texts with the annotation.** For measuring the effectiveness of the disambiguation heuristics, three widely used performance measures were used:

- *Precision* is the proportion of correctly detected references to all detected references (correct and false hits, eq. 3.1) (Oakes 1998).
- *Recall* measures the proportion of correctly detected references to all references in the annotation (both correct and missed hits, eq. 3.2) (Oakes 1998).
- The *F-Score* merges precision and recall into a single efficiency measure by calculating the harmonic mean of the two (van Rijsbergen 1979, eq. 3.3<sup>8</sup>).

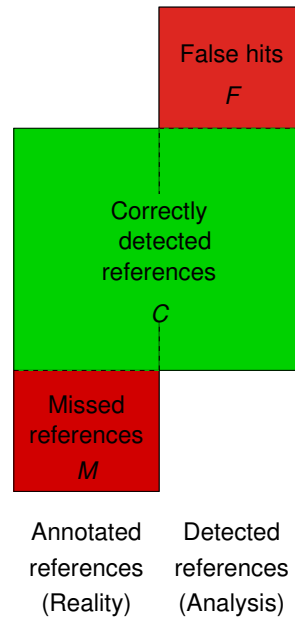


Figure 3.1: Correctly detected references  $C$ , false hits  $F$  and missed references  $M$  of an analysed text.

Figure 3.1 clarifies the concepts of correctly detected references  $C$ , missed references  $M$  and false hits  $F$ .

$$\text{Precision } P = \frac{C}{C + F} \quad (3.1)$$

$$\text{Recall } R = \frac{C}{C + M} \quad (3.2)$$

$$\text{F-Score } F = \frac{2PR}{P + R} \quad (3.3)$$

<sup>8</sup>van Rijsbergen (1979) introduces the F-measure as an error function:  $E = 1 - \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}$ , where  $\alpha$  is a weight factor with  $0 \leq \alpha \leq 1$ . The F-measure as an effectiveness measure is defined as  $F = 1 - E$  (Makhoul et al. 1999). To equally weigh recall and precision,  $\alpha$  is set to 0.5, resulting in  $F = \frac{1}{\frac{1}{2P} + \frac{1}{2R}} =$

$\frac{1}{\frac{2R+2P}{2P \cdot 2R}} = \frac{2PR}{P + R}$ , as given in eq. 3.3.

## 3.2 Geo-Coding heuristics

### 3.2.1 Context extraction

Most news articles have a geographical focus: Both if they report on a specific event (*bomb explosion in Baghdad, plane crash in Russia, elections in Afghanistan*) or on a broader topic (*German pension scheme since the 1890s, The US role in the Middle East*), one or a few countries can be extracted on which the article focuses on. In the following, these countries in focus will be referred to as the *geo-context* of a text.

The geo-context will be used for intelligently querying the database with the shallow-deep parsing approach described in section 3.2.2. Furthermore, the geo-context will be used by the context-based triggering heuristic (section 3.2.4). Consequently, a reliable geo-context extraction is a key part of the place name filtering and disambiguation process.

For this work, three indicators for the inference of a geo-context were implemented:

- Place of publishing, which can be set explicitly.
- Place of writing, which is often contained at the beginning of a text.
- Shallow parsing of text.

#### Place of publishing

While the big news agencies and media companies report on news from all over the world, smaller news sources often have a focus on regional and local news. Usually, these sources concentrate on the region they are published in.<sup>9</sup>

In order to be able to make use of the local focus of news sources, a list of newspapers and their main focus was created. The granularity of information stored for a news source depends on its ‘locality’: news agencies with a global focus are not assigned a focus at all, local newspapers have entries for all granularities, namely continent, country, and region (see also table 3.2). Multiple entries for countries and regions are possible.

The list of news source coverage was created manually. Indicators for estimating the locality were either the news sources’ subtitles (like *East Anglican Daily Times - The morning newspaper for Suffolk and Essex*) or information found on the sources’ web pages (e.g. category for a specific region on the website). If no additional information could be found, no regional geo-context was set.

---

<sup>9</sup>An exception are exile newspapers: These papers often report on a country lacking freedom of press (e.g. Cuba) while they are published in a country where many of the exiles live (e.g. the US). See Brown Jr. and Botero (1997) for an overview of Cuban exile newspapers.

News source	Focus	Continent	Country	Region
Reuters	Global	–	–	–
Africa Daily	Continent	af	–	–
Baltic Times	Multinational	eu	EE, LT, LV	–
Le Monde	National	eu	FR	–
Frankenpost	Local	eu	DE	328 (Bavaria)

Table 3.2: Coverage of different news sources. The continent and country codes are ISO2 codes, the region code is a unique number representing an administrative unit.

It would also be possible to evaluate the geo-context of all articles from one news source. If it is found that a news source mainly reports on one country, this country could be set in the publishing list. An example: According to Bruno Pouliquen, an analysis of the geo-context showed that 53% of all articles published by the *Times of Malta* report on Malta (personal conversation, August 13, 2004); in that case, Malta would be automatically set as the main country.

For making use of the coverage list, the news source must be specified when analysing a text. In order to add the associated country<sup>10</sup> to the geo-context, it has to be backed in the shallow parsing of the text (see also section 3.2.2). That means, the country the news source is published in has to occur at least once more in the text to add it to the geo-context.

### Place of writing

Another indicator for the geo-context is the place where a text was written: Especially news agencies and bigger media companies have correspondents in many countries, which then report about these countries. Frequently, the place where the article was written is mentioned at the beginning of the text. Examples including the place of writing are given in table 3.3.

To extract this place of writing from the text, we analyse the first 50 characters of an article. Therefore, for every word at the beginning of a text a query in the gazetteer is performed; this query is more or less a tiny geo-parsing for the initial words of the text. If a place is found, the country it lies in is added to the geo-context of the text.

---

<sup>10</sup>Since for this work only the geo-context on a country level was used, only the country information is used. Further work needing a finer-grained geo-context can also evaluate the regional information.

Beginning of article	Extracted country
WASHINGTON (CNN) – [...]	USA
(Lagos) July 1, 2004 [...]	Nigeria
By Luke Baker BAGHDAD, Jul 1 (Reuters) - [...]	Iraq
Presov/Bratislava, August 6 (TASR-SLOVAKIA) - [...]	Slovakia
<i>Αθήνα</i> , 9, <i>Αύγουστος</i> 2004 - [...]	Greece

Table 3.3: A selection of newswire texts mentioning the place of writing at the beginning. [...] represents the beginning of the first sentence of the article.

### Shallow parsing of the text

Since publishing information does not have to be set and the place of writing may not be contained at the beginning of a text, a parsing of the complete text is the only reliable way to infer the geo-context of a text. For this work, a shallow parsing of the text is performed, i.e. only the most important places are queried (see also section 3.2.2).

In order to add a country to the geo-context, at least three references to it<sup>11</sup> must be contained in the analysed text. Alternatively, a country is added to the geo-context if its references make up at least 50% of all references in a text.

### 3.2.2 Shallow-deep parsing

As Pouliquen et al. (2004) indicate, large gazetteers add ambiguity: since a large gazetteer contains more entries, it is more likely that it contains homonyms to other words in natural language. Furthermore, there are more alternatives sharing the same place name. To fight that problem of ambiguity, Pouliquen et al. (2004) artificially reduced the size of the database; this approach, however, can never detect places which have been deleted from the database, even if they would be correct hits. Therefore, a high precision ratio (less false hits) is paid dearly with a low recall (less correctly detected hits).

If it would be possible to a priori filter references which are false hits in that particular context, while they still can be detected if they are correct hits, both the precision and the recall would benefit. In other words, an intelligent filtering of insignificant places and hence a context-dependent reduction of the search space is thought to improve the results significantly.

<sup>11</sup>A reference to a country is given if either the country name itself or a region or place name lying in that country occurs in the text. Hence, *France*, *Île de France*, and *Paris* are references to France.

This filtering issue is solved by looking for places in a two-pass process:

- In a first pass, a shallow parsing of the database is performed. That means, only places are queried whose importance (represented by a place class attribute in the gazetteer) is greater than or equal to a certain threshold. The places found in this first pass are then used to infer a geo-context of the text, as described in section 3.2.1. In the example in figure 3.2, the countries denoted by red and blue areas represent the geo-context.
- In a second pass, a deep query of the database is performed, i.e. all places with a lower importance than the threshold are taken into account. However, the search is limited to places in countries which were found to be in the geo-context of the text in pass one. In the example in figure 3.2, the deep search would be limited to the red and blue hatched areas.

This two-pass approach has the advantage that in the first pass it limits ambiguity by searching only a subset of the gazetteer. The second pass, however, ensures that also small places can be detected. See section 3.3.1 for an analysis of the performance of this shallow-deep parsing process with different shallow-deep thresholds.

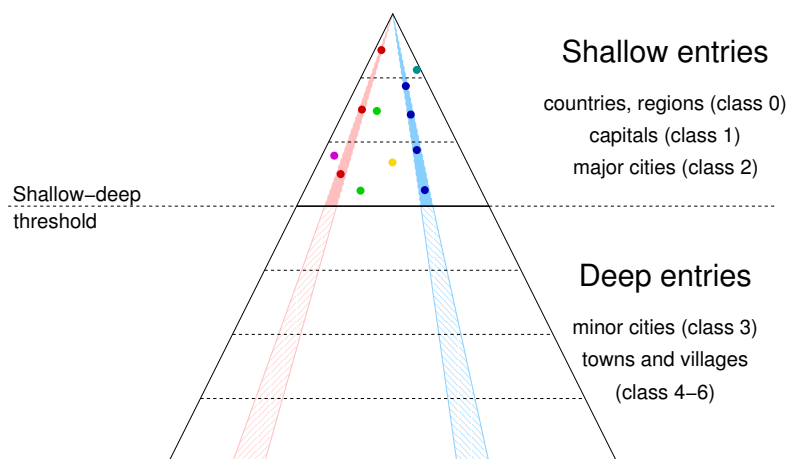


Figure 3.2: Example clarifying the shallow-deep parsing:

In a first pass, the shallow parsing returns 12 hits, which are represented by dots. Different dot colours denote different countries. For two countries (red and blue), at least three places are detected, so they are added to the geo-context.

The second pass now searches the deep entries, but restricts to the countries which are in the geo-context, i.e. only the red and blue hatched areas are queried.

### 3.2.3 Place name filtering

As indicated in section 2.2.1, many place names are homonymic with other words in natural language, or with persons' names. In this work, four heuristics are used to tackle this problem and to filter out false hits which are no place names:

- A person name detection heuristic by Bruno Pouliquen (PN module).
- A manually created list of around 100 names of important persons, mainly politicians (VIP list).
- An automatically generated and manually enhanced list of words frequently used in natural language. These words might also refer to places, but in most cases are function words. Therefore this list is called geo-stop-list.
- A heuristic which scans the context of a word for place triggering words.

If a potential place name is detected in one of the approaches above, it is given a weight according to the weighing parameters given in table 3.4 on page 34. That means, if a potential place name (e.g. *Annan*) is found to be a person's name, its probability that it is really a place becomes lower. However, the hit is not completely sorted out, since it was found in the database and thus refers to a place: Though *Annan* will often refer to the UN Secretary-General, it might still stand for the town *Annan* in Scotland.

In the following, the four approaches described above will be presented in more detail.

#### Heuristical person name detection

Bruno Pouliquen created a person name detector, which is also used for the JRC's *Top stories* application, where it extracts a list of the "Persons of the day" out of all articles analysed by the application (see also Pouliquen, Steinberger and Ignat (2004)).

It uses a heuristic which compares potential person names with a list of words often occurring in the context of persons, like *president*, *Dr.*, or *said*. Generally, it assumes that a name consists of two subsequent upper case words (Jacques Chirac, Tony Blair), but it also detects names containing conjunctions (Charles *de* Gaulle, Richard *von* Weizsäcker). Middle name initials (George W. Bush, George W Bush) are also supported. References to persons consisting solely by the second name (Bush, Chirac, Blair) are currently not detected.

The hitherto unpublished heuristic is encapsulated in a Perl module, and could be used without further adaptations. If the person name detector classifies a potential place

as a person name, the weight for this word is decreased by 20, i.e. the probability that the potential place is really a location becomes lower (see also table 3.4 on page 34).

### VIP lists

An alternative to the just mentioned heuristical person name detection is to create a list of frequently occurring person names, and to check whether these names are contained in the text to be analysed. Zipf's law shows that the most frequent words in a language occur by far more often than the less frequent ones,<sup>12</sup> and therefore a relatively short list of the most frequent entries is capable of covering many occurrences of a type; Rivero Rojas (1999) showed that Zipf's law also holds for the frequency of scientific authors' references in articles, and it can be assumed that a similar correlation also exists for the frequency of person names in newswire texts.

Therefore, it seems promising to compile a list of person names which occur frequently in newswire texts: even with a short list a relatively high number of person names can be detected.

The VIP list generated for this work focuses on VIPs from politics and includes the most important politicians since World War II, the UN Secretaries-General, and the current EU commissioners. Well-known persons from other domains like history, sports, entertainment etc. are not yet included in the list, but could be added later if required. Furthermore, it would be possible to automatically update the VIP lists by adjusting them with the "Persons of the day" generated by the *Top stories* application described in Pouliquen, Steinberger and Ignat (2004), which daily compiles a list of the persons currently in the media. To keep the list simple, this work just uses a manually compiled list containing around 100 names of VIPs from politics.

To allow for language specific variations in spelling and character set, several lists were created. The general list contains all names in their original diction, including all accentuation, and in the original character set. In this list, for instance, Gorbachev is stored in Cyrillic (Горбачёв).

In addition to this general list, one list for each language was created, which contains language specific spelling variations and transcriptions. To follow the example from above, the English list would contain the entry *Gorbachev*, the French one *Gorbachev*, and the

---

<sup>12</sup>Zipf's law states that there is a "constant relationship between the frequency of a word in a corpus [of natural language] and its rank" (Zipf 1949, p. 54, cited in Rivero Rojas 1999). Specifically, Zipf showed that the most frequent word in a corpus occurs 10 times more often than the 10th most frequent word, and 100 times more often than the word with rank 100.

German one *Gorbatschow*.

The lists are stored as hashes in Perl modules, and can easily be maintained and enlarged. For the analysis, the potential places in the texts are compared to the entries in the VIP lists; if a match is found, the weight for this potential place is decreased by 20.

### **Geo-stop-lists**

Following the motivation behind the VIP lists (see above), a list containing homonyms from natural language was used to filter out potential places which are usually words in natural language.

This idea was brought up in Pouliquen et al. (2004), where the lists are called “geo-stop-lists”. In this work - and as opposed to Pouliquen et al. (2004) - a word contained in the list is not sorted out completely (“stopped”), but is given a negative weight; consequently, other indicators might still outweigh this factor. Anyhow, as in Pouliquen et al. (2004), the lists containing frequent words which are usually not geographical references will be called “geo-stop-lists” in the following.

As a basis, the geo-stop-lists used by Pouliquen et al. (2004) were taken. They were automatically generated by analysing a set of test texts and adding those words to the geo-stop-lists which mainly lead to wrong hits. The stop-words which are common to several languages are stored in a multi-lingual geo-stop-list, the specific ones in a language-specific list. This basic list was manually extended by adding those most frequent function words of each language (as given in 2.1) to the list which are also place names in the gazetteer.

In total, each of the language-specific lists comprises between 30 and 50 entries. The lists are stored as hashes in Perl modules, and can easily be maintained and enlarged. If, in the analysis, a potential place is contained in the geo-stop-list, its weight is decreased 30 (see also table 3.4).

### **Place triggering**

By now, only place filtering heuristics which aim at sorting out false positives have been discussed. The place triggering heuristic introduced now, on the other hand, looks for indications that a word is really a place name.

The idea behind the place triggering heuristic is that many geographical references - especially the ones representing smaller places like villages - have a word indicating the place type in their surrounding. An example: If no context is given, it is unclear what “Chakpota” is - it might be a person’s name, a place name, or no named entity at all. When looking at its context - “in Chakpota village, 100 miles north of Dhaka” -

it becomes clear that Chakpota is a place (namely a village), and an indication is given about its geographical position. Therefore, the idea is to look for words triggering places in the surroundings of a potential location.

In order to be able to detect place triggering words, a list containing around 15 of such words was created for each language. To be able to differentiate between different types of places, the entries are assigned to one of five place types: populated place, waters, island, administrative unit, and landmark. This grouping into types of places might help to assign a reference to its real-world counterpart in the subsequent disambiguation phase; however, for this work this has not been implemented. As for the VIP and geo-stop-lists, the data is stored in hashes in Perl modules.

In the analysis, the surrounding ten words of each potential place (each five to the left and to the right) are compared to the place triggering list. If a word indicating a place is found, a positive value is added to the weight of the potential place. The weight depends on the distance (in words) between potential place and place triggering word - the maximal value is 50 which is added if the place indicator is right beside the potential place. In this work, a quadratic decrease function is used.

### 3.2.4 Place name disambiguation

In the place name filtering phase, potential places which indeed are persons' names or other words in natural language were sorted out (see also 2.2.2). In other words, after the filtering phase we act on the assumption that a potential place is really a place. However, it is not known to which place alternative a reference refers to.

To solve this issue, several heuristics are used to assign a potential place name in the text to its most probable real-world counterpart:

- *The relative importance of a place*

In general it can be assumed that bigger and more important places (capitals, major cities) are more likely to occur in a text than small towns or villages. Hence, supposedly a place name is likely to stand for an alternative with high importance (i.e. a low class in the database).

- *Context-based triggering*

News texts often have a geographical context. If a place alternative is in the country a text is about (i.e. in the geo-context), it is more probable to refer to the actual place than an alternative which lies in an unrelated country.

- *Comparison of a location to other places in the text*

Even if a country is not in the *geo-context* of a news article, it might still appear as a word in the *text*. Again, the co-occurrence of a place alternative and its country in a text makes this combination more probable than other, unrelated ones.

- *The distance of locations from the event*

News texts often have a place of occurrence, an “epicentre”. It is more likely that a place name refers to an alternative which is closer to that epicentre than to ones which are further away.

In this study, two variations of this heuristic are used: one calculates the minimal distance of an alternative to one of the unambiguous places, the other one calculates the average distance to all unambiguous places.

Each of the place alternatives is analysed with these heuristics, and according to the result (class, distance etc.), a weighing parameter as given in table 3.4 is added to the alternative’s weight. If *Paris* occurs in the text, the relative-importance-heuristic will add much weight to the alternative standing for the French capital. However, in a text about Bourbon whiskey, *Paris* would more likely refer to Paris, KY (USA).<sup>13</sup> This would be reflected in higher weights from the context-based and distance-based heuristics.

### Relative importance of places

Probably the most straightforward heuristic to guess a place alternative is to look at its importance. It can be assumed that bigger and more important places (capitals, major cities) occur more often in a text than small towns or villages, i.e. their occurrence is more probable. Consequently, locations with higher importance (i.e. with a lower class) should be given more weight than ones with higher classes.

Depending on their place class, each alternative is given a weight between 5 (small villages) and 80 (countries, administrative units, countries’ capitals, major cities). See table 3.4 for a complete list of weights.

### Context-based triggering

As shown in section 3.2.1, most newswire texts have a geographical focus, which is defined as the *geo-context* of a text. Therefore it can be assumed that a place name refers more likely to a place alternative lying in a country in the *geo-context*, than to an alternative lying in a country which was never mentioned in the text.

---

<sup>13</sup>Paris, Kentucky is the capital of Bourbon County, after which the whiskey is named (Wikipedia 2004a).

To come back to the example from above: in a text about Bourbon whiskey from Kentucky and Tennessee (geo-context set to the US), which contains the geographical reference *Paris*, the alternative Paris, KY (USA) would be given a positive weight, since it lies in a country in context.

The context-based triggering heuristic makes use of that assumption: if a place alternative lies in a country which is in the geo-context of the text analysed, a positive value of 100 is added to its weight (see also table 3.4).

### **Comparison of a location to other place names in the text**

A place can hint to a specific place alternative even if the country it lies in is not in the geo-context - for example because there are too few occurrences to add the country to the geo-context. By coupling the weight to the number of occurrences of a country, this text-based comparison is more fine-grained than a simple context-based triggering.

The locations hinting towards one alternative often occur near another, ambiguous place: they may specify the region an ambiguous place lies in (*Boston, Massachusetts; Freiburg im Breisgau*) or give evidence of vicinity (*in Chakpota village, 100 miles north of Dhaka; Raf-Raf, a small coastal town between Tunis and Bizerte*).

It is mostly the case that these vicinal places lie in the same country, and therefore a place alternative can simply be inferred from an unambiguous location in its proximity. Therefore, place alternatives which are in the same country as another, unambiguous place in the text, are more probable than alternatives which have no such reference in the text. More complex approaches like spatial queries, as for example in Bilhaut et al. (2003), or inherent part-of relations, as in Leidner, Sinclair and Webber (2003), are not used in this work, because the necessary information is not contained in the gazetteer.

For each unambiguous place which lies in the same country as the place alternative, a value of 10 is added to the weight of the place alternative (see also table 3.4).

### **Distance of locations from the event**

It can be assumed that place alternatives, which are closer to other geographical references in the text, are more likely than alternatives which are further apart. Therefore, the kilometric distance of the place alternatives to unambiguous places in the text is taken into account.

Two heuristics making use of the distance of locations are applied: the first one, referred to as “minimal distance heuristic”, calculates the distance of an alternative to the unambiguous place which is nearest to it. The second heuristic, which from now on will be

called “average distance heuristic”, calculates the average distance of a place alternative to all unambiguous places in the text.

Both heuristics use an algorithm proposed by Sinnott (1984) for computing the distance of two locations on earth. The calculated distances are then weighed by using a modified *arccot* function. More details on these issues are presented in the following.

**Calculating the distance of two locations on earth.** In the plain, the shortest distance of two points is a straight line, their distance  $d$  is defined as  $d = \sqrt{(\Delta x)^2 + (\Delta y)^2}$ . Since the earth is round, this straightforward approach cannot be applied: The shortest way between these two points would lead through the bowels of the earth. Instead, the shortest distance of two places following the earth’s *surface* is to be found.

With the simplifying assumption that the earth is a perfectly round sphere,<sup>14</sup> spherical geometry can be applied. Following a suggestion by Chamberlain (1997), the earth radius is inferred from the definition of a nautical mile and set to  $R = 6366.71km$ .<sup>15</sup>

Bronstein et al. (1999, eq. 3.190b) define the distance of two places  $P1(lat1, lon1)$  and  $P2(lat2, lon2)$  on earth with

$$d = arccos [sin(lat1)sin(lat2) + cos(lat1)cos(lat2)cos(\Delta lon)] \cdot \frac{\pi R}{180} \quad (3.4)$$

Though this definition is mathematically exact, Sinnott (1984) argues that it is vulnerable to rounding errors for small distances, and proposes a mathematically equivalent formula, which was also used for this work (Sinnott 1984, p. 159, cited in Chamberlain 1997):

$$d = 2 \cdot R \cdot arcsin [min(1, \sqrt{a})] \\ \text{with } a = \left( sin\left(\frac{\Delta lat}{2}\right) \right)^2 + cos(lat1) \cdot cos(lat2) \cdot \left( sin\left(\frac{\Delta lon}{2}\right) \right)^2 \quad (3.5)$$

**Weighing the distance.** For weighing the distances of places, a weight function with the following characteristics was looked for:

1. The function should clearly favour places which are near to each other. However, for small distances the function should only decrease slightly: whether two places are 50 or 60 kilometres away should not affect the result strongly.

<sup>14</sup>The earth is oblate at the poles, i.e. the diameter between the poles is slightly smaller than the diameter at the equator. When assuming that the earth is a perfect sphere, the results at these extreme points will slightly deviate. But since we are only interested in an estimation of distances rather than exact values, this is acceptable.

<sup>15</sup>One nautical mile is defined as “one minute of arc of a great circle of the earth” (Chamberlain 1997). Taking the currently accepted value of a nautical mile, 1.852 km, the earth radius can be defined as  $R = \frac{360 \cdot 60 \cdot 1.852}{2\pi} km = 6366.71km$ .

2. Similarly, it does not make much difference if two places are 5,000 or 6,000 kilometres away. Therefore the decrease of the function for large distances should also be slight.
3. Empirical investigation showed that for places with a distance of more than 500 kilometres no correlation can be seen which is due to the distance of these places. For distances below 200 kilometres such mutual relationships are apparent. Therefore the function should have its steepest descent between  $200 < x < 500$ .

The arc cotangent function  $\operatorname{arccot}(x)$ , as defined by Bronstein et al. (1999, eq. 2.145, see also figure 3.3a),<sup>16</sup> naturally fulfils the first two prerequisites: As a continuously decreasing function with  $\lim_{x \rightarrow -\infty} \operatorname{arccot}(x) = \pi$  and  $\lim_{x \rightarrow +\infty} \operatorname{arccot}(x) = 0$  neither very small nor very big  $x$ -values are weighed very differently. The third prerequisite can be fulfilled by moving the inflexion point - i.e. the point with the steepest descent - from  $x = 0$  to  $x = 300$ . By stretching the function by a factor of 100 and normalising it so that  $w(0) = 1$ , we get the weight function  $w(x)$ , as given in eq. 3.6 and shown in figure 3.3b:

$$w(x) = \frac{1}{\operatorname{arccot}(-3)} \cdot \operatorname{arccot}\left(\frac{x - 300}{100}\right) \quad (3.6)$$

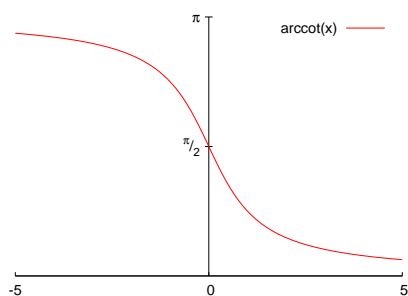


Figure 3.3a: The arc cotangent function  $\operatorname{arccot}(x)$

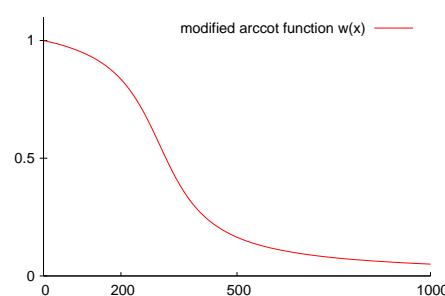


Figure 3.3b: The modified arc cotangent function  $w(x)$ , as given in eq. 3.6

<sup>16</sup>Bronstein et al. (1999) define  $\operatorname{arccot}(x)$  as the inverse function of  $\cot(x)$  with domain  $[0 \leq x \leq \pi]$ , which is equivalent to  $\frac{\pi}{2} - \arctan(x)$  (Bronstein et al. 1999, p. 82ff.). According to this definition,  $\operatorname{arccot}(x)$  is a continuously decreasing function without discontinuities, as shown in figure 3.3a. In contrast, Weisstein (1999a) defines  $\operatorname{arccot}(x) = \arctan(\frac{1}{x})$  which has a discontinuity at  $x = 0$ . The latter definition, however, violates Weisstein's own definition of an inverse function  $f^{-1}(x)$  as being a reflection of  $f(x)$  about the line  $y = x$  (Weisstein 1999b). During this work, the definition by Bronstein et al. (1999) will be used.

### 3.2.5 Weighing the heuristics

In this work, four place filtering and five place disambiguation heuristics are used in order to detect locations and map them to their real-world counterparts. When used in combination, these heuristics are weighted by additive weight functions.

Two weight functions will be used:

1. The first weight function combines four place *filtering* heuristics ( $N_F = 4$ ). This weight function aims to quantify if a potential place  $PP$  is estimated to be a place or not.

$$W_{PP} = b + \sum_{n=1}^{N_F} w_n \cdot v_n \quad (3.7)$$

2. The second weight function combines five place *disambiguation* heuristics ( $N_D = 5$ ). A weighing is performed for each place alternative  $A$ . The alternative with the highest weight is assumed to be the most probable place.

$$W_A = \sum_{n=1}^{N_D} w_n \cdot v_n \quad (3.8)$$

In both weight functions,  $w_n$  is the weight factor for heuristic  $n$ ,  $v_n$  is the value of heuristic  $n$ . For the heuristics with boolean results, this value  $v_n$  will be either 1 if their characteristic is distinct, or 0 if not. For heuristics with a continuous range,  $v_n$  is normalised to values between 0 and 1. The disambiguation heuristic comparing a place alternative to other references (“text” heuristic) sets  $v_n$  to the number of unambiguous places in a text which lie in the same country as the alternative.

For the place filtering weight function, the bias  $b$  is set to 10. A potential place is assumed to be a place if the result of the weight function is greater than 0. That means, if none of the place filtering heuristics responded (leading to terms of 0), a potential place will be assumed to be a place, since the bias is set greater than 0. If  $W_{PP} < 0$ , the potential place is assumed *not* to refer to a place, but instead to a person’s name or another word in natural language. In other words, the potential place has been sorted out.

For the place disambiguation heuristics, no bias is needed, because the alternatives are compared to each other, and a bias would only change the absolute values, but not the actual rank of the alternatives. Similarly, a threshold is dispensable: the alternative with the highest absolute value  $W_A$  is considered as the most probable one.

The weight factors  $w_n$  of the heuristics were set manually: The development sets (i.e. the texts in the *Turkey* and *Chechnya* clusters) were analysed with the heuristics described in section 3.2. The results of the analysis were then evaluated and the weights for each heuristic were manually adjusted so that as many references as possible were correctly detected, and so that the number of false hits was minimal. Table 3.4 shows an overview of the weights and possible values from the heuristics used in this work.

	<b>Heuristic</b>	<b><math>w_n</math></b>	<b><math>v_n</math></b>	
Place filtering heuristics (weight func. 3.7)	Heuristical place name detector (PN module)	-20	0 or 1	
	VIP list	-20	0 or 1	
	Geo-stop-list	-30	0 or 1	
	Place triggering	50	[0, 1]	
Place disambiguation heuristics (weight function 3.8)	Place's country in geo-context	100	0 or 1	
	Place's country in text	10	$N^a$	
	Place class	0, 1 or 2	80	0 or 1
	"	3	30	
	"	4	20	
	"	5	10	
	"	6	5	
	Kilometric distance (avg.)		20	]0, 1]
Kilometric distance (min.)		30	]0, 1]	

<sup>a</sup> $N$  = number of unambiguous places in a text which lie in the same country as the alternative

Table 3.4: Weights  $w$  and possible values  $v$  for place name filtering and disambiguation heuristics.

### 3.3 Results

This section presents the results of applying the heuristics to the development and test sets introduced in section 3.1.4. Each of the following subsections will focus on one aspect (e.g. place filtering heuristics, language etc.), while the other variables do not change. In section 3.3.1 the effect of the threshold value for the shallow-deep parsing approach, which was introduced in section 3.2.2, will be examined. The two following sections analyse the impact of each place filtering and place disambiguation heuristic on the overall result. In section 3.3.4 it will be evaluated how the heuristics work for different languages, while in section 3.3.5 differences between the topics will be investigated. Section 3.3.6 looks at the running time of each heuristic and proposes a set of heuristics producing valuable results in an adequate time. The overall results presented will be summarised in section 3.3.7.

Each of the result sections is structured as follows: At first, the overall results - which are represented by the F-Score - are presented. Then, the results are looked at in more detail by analysing recall and precision.

**How to read the results.** In the following sections, several figures showing F-Score, recall and precision of a certain combination of parameters will be shown (figures 3.4 to 3.8). Each of these figures contains several bars:

- The *single-hatched bars* show the values for the analysis with the heuristics used in Pouliquen et al. (2004). These results are the baseline of the analysis.
- The *cross-hatched bars* show the values for the analysis with heuristics from this work, where one or more additional options (e.g. a part of the heuristics) are not used.
- The *solid bars* also show the results for the analysis with heuristics from this work, but here the additional options are indeed activated. Hence, the difference between cross-hatched and solid bars indirectly represents the additional impact of the additional option.

Furthermore, most of the figures show results for two different test groups:

- The *yellow bars* show the results for the analysis of the development sets, i.e. the texts of the *Turkey* and *Chechnya* clusters.
- The *blue bars* show the results for the analysis of the test sets, i.e. the texts of the *Iran* and the *Reagan* clusters.

Since many F-Score, recall and precision values are similar, and therefore hard to discriminate in the graphs, the actual values are given in tables in the appendix (tables A.2 to A.6 on pages 70 to 74).

### 3.3.1 Performance of the shallow deep parsing

**Overall results.** As figure 3.4 shows, the shallow-deep parsing leads to significantly better results than the heuristics used by Pouliquen et al. (2004). The best overall results are observed for a threshold value of 2. A threshold of 2 means that only countries, administrative units, countries' capitals and major cities are taken into account in the first pass. With the exception of threshold 1, larger threshold values lead to lower F-Scores.

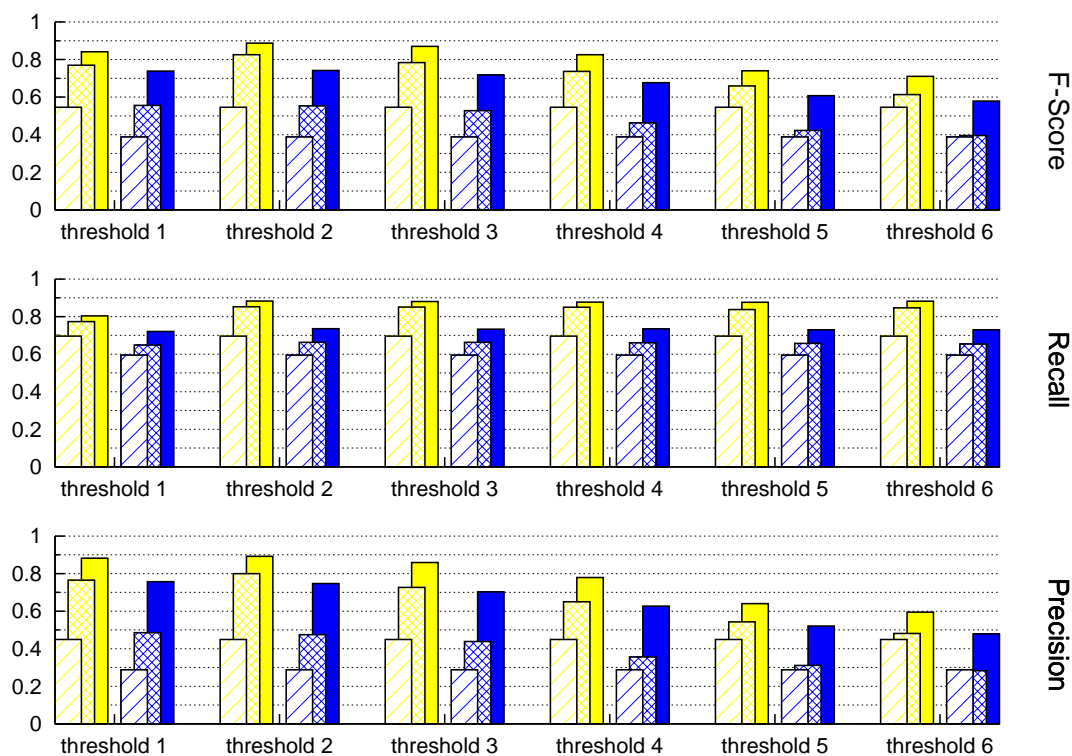
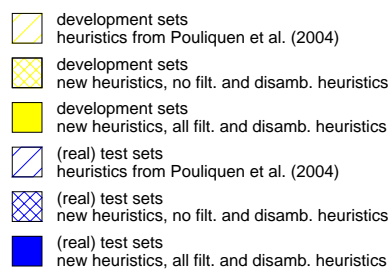


Figure 3.4: F-Score, recall and precision for different shallow-deep parsing thresholds. The single-hatched bars show the results from the heuristics used in Pouliquen et al. (2004). The cross-hatched and solid bars use the extended gazetteer from this work, with no respectively all heuristics been used.



Reasons therefore will be given in the analysis of recall and precision values.

**Recall.** For a threshold of 2 or greater the recall values are more or less constant. That means, that even a rather low threshold value of 2 returns all relevant places. If the threshold value is increased more, the recall values decrease slightly because more place alternatives are returned, which may be wrongly accounted for the place which is referred to in the text. Consequently, the number of correct hits decreases, leading to lower recall values - see also table A.2. If, on the other hand, the threshold is set to a value of 1, the recall decreases significantly. This is due to the fact that the geo-context cannot be inferred properly, which means that many places will not be returned in the deep parsing step. Hence, less geographical references can be detected, resulting in a lower recall.

**Precision.** As can be seen in figure 3.4, threshold values of 1 and 2 lead to the highest precision values. Bigger threshold values clearly affect the precision values negatively. The reason is that for big threshold values more potential places will be returned, which then might not be sorted out correctly by the disambiguation heuristics. This results in a bigger number of false hits, which lowers the precision ratio. The minimal threshold value 1 tends to produce the lowest number of false hits, but since it returns less correct hits, the precision is at the same level as for threshold 2.

### 3.3.2 Performance of the place filtering heuristics

For the evaluation of the place filtering heuristics, two analyses were performed: At first, the texts were analysed with the various place filtering heuristics while *none* of the place disambiguation heuristics was used in the subsequent second step. This test run is represented by the cross-hatched bars in figure 3.5. In the second analysis, *all* place disambiguation heuristics were used (solid bars in figure 3.5). Both analyses use a shallow-deep threshold value of 2, which was found to perform best in section 3.3.1. For comparison reasons, the performance of the baseline algorithm presented in Pouliquen et al. (2004) is also shown (single-hatched bars).

**Overall results.** As figure 3.5 shows, the newly developed place filtering heuristics clearly outperform the ones presented in Pouliquen et al. (2004). Improvements can be seen in all aspects (compare table A.3 on page 71): more correct hits are detected (between 8.0% and 27.2% more) while the number of missed hits decreases (between 11.7% and 61.5%). Furthermore, there is a significant drop in the number of false hits (between 50.2% and 87.5%). Each of these factors contributes to significantly higher F-Scores. A combination of all place filtering and disambiguation heuristics leads to the best overall results: the F-Score for the test sets rises from 0.388 (results for approach from Pouliquen

et al. (2004)) to 0.741, which is an increase of 91.0%.

It is apparent that both the baseline heuristics by Pouliquen et al. (2004) and the improved heuristics from this work show better performance for the development sets (yellow bars in figure 3.5) than for the test sets (blue bars). The cause for this difference will be discussed in detail in section 3.3.5. In any case, for both development sets and test sets the new heuristics clearly outperform the ones used by Pouliquen et al. (2004).

The heuristical place name detector (PN module), the VIP lists, and the geo-stop-lists filter out many of the false positives and therefore each heuristic improves the overall result. The fact that the PN module and the VIP lists show their strengths in the test sets, while the geo-stop-lists have advantages in the development sets suggests that the

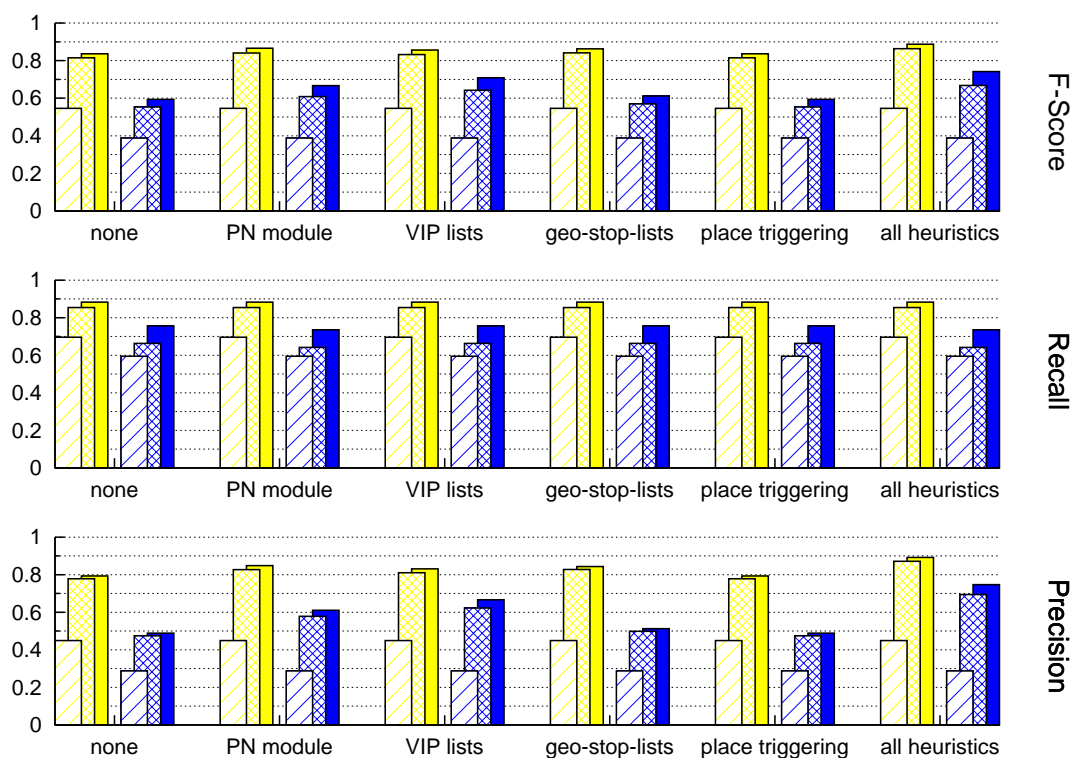
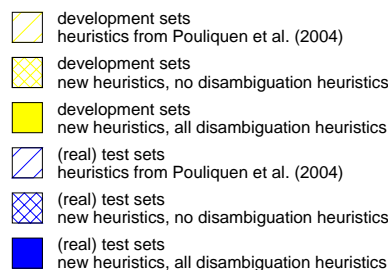


Figure 3.5: F-Score, recall and precision for the place name filtering heuristics with all (solid bars) and no disambiguation heuristics (cross-hatched bars) used. Shallow-deep threshold is 2. The single-hatched bars represent the results from the heuristics used by Pouliquen et al. (2004).



performance of these heuristics depends on the texts analysed. Consequently, in order to account for all the different aspects, only a combination of different heuristics can lead to optimal results. This assumption is backed by the results of the analysis which uses all heuristics in combination: it leads to the best overall results.

The place triggering heuristic, however, could not improve the result at all. A possible explanation is that it backs the “a-potential-place-is-really-a-place” assumption for words which are already known to be a place - in this case the information is redundant.<sup>17</sup> The case where the place triggering heuristic might reverse a false hit by another heuristic - e.g. that *Annan* is found to be a person’s name by the heuristical place name detector, though it is in the context *town of Annan* - seems not to be apparent in the development and test sets.

**Recall.** The place filtering heuristics have only little effect on the recall values. Since it is the disambiguation heuristics which assign a place alternative to a reference, the filtering heuristics do not have an impact on the number of correctly detected references; since also the number of missed references is more or less constant, the recall does not change either. However, when applied to the test sets, the PN module filters out around 20 correct hits, which then become missed hits. In consequence, the recall value for the PN module and for the combination of all heuristics is lower than for the other heuristics. In this sense, the PN module is counter-productive.

**Precision.** The motivation for using place filtering heuristics was to filter out false positives - therefore it was assumed that they have a positive impact on the precision value. Indeed, the number of false positives decreases significantly: for the test sets, a combination of all heuristics filters out 83.1% of the false positives counted for the baseline algorithm by Pouliquen et al. (2004). The PN module, the VIP lists, and the geo-stop-lists fulfil this expectation and lead to precision values of around 0.80 for the development sets and between 0.48 and 0.67 for the test sets. Compared to the baseline algorithm, that corresponds to a relative increase between 73.3% and 131.3%. If all filtering heuristics are used in combination, the precision value for the test sets even rises by 159.4%, or from 0.288 to 0.747.

---

<sup>17</sup>Each word in a text is estimated a potential place name if it is contained in the gazetteer, because a gazetteer only contains place names. Hence, an additional triggering of places which are known to be places can be called redundant.

### 3.3.3 Performance of the place disambiguation heuristics

Two analyses were performed to measure the impact of each place disambiguation heuristic: in a first analysis, the test texts were analysed without using the place filtering heuristics (cross-hatched bars in figure 3.6), the second analysis uses all the place filtering heuristics (solid bars). In both analyses, a shallow-deep threshold of 2 was used. Again, the performance of the algorithm used in Pouliquen et al. (2004) is included in figure 3.6 for comparison reasons (single-hatched bars).

**Overall Results.** As figure 3.6 shows, the F-Score for each single disambiguation heuristic is significantly higher than the F-Score by the baseline algorithm from Pouliquen

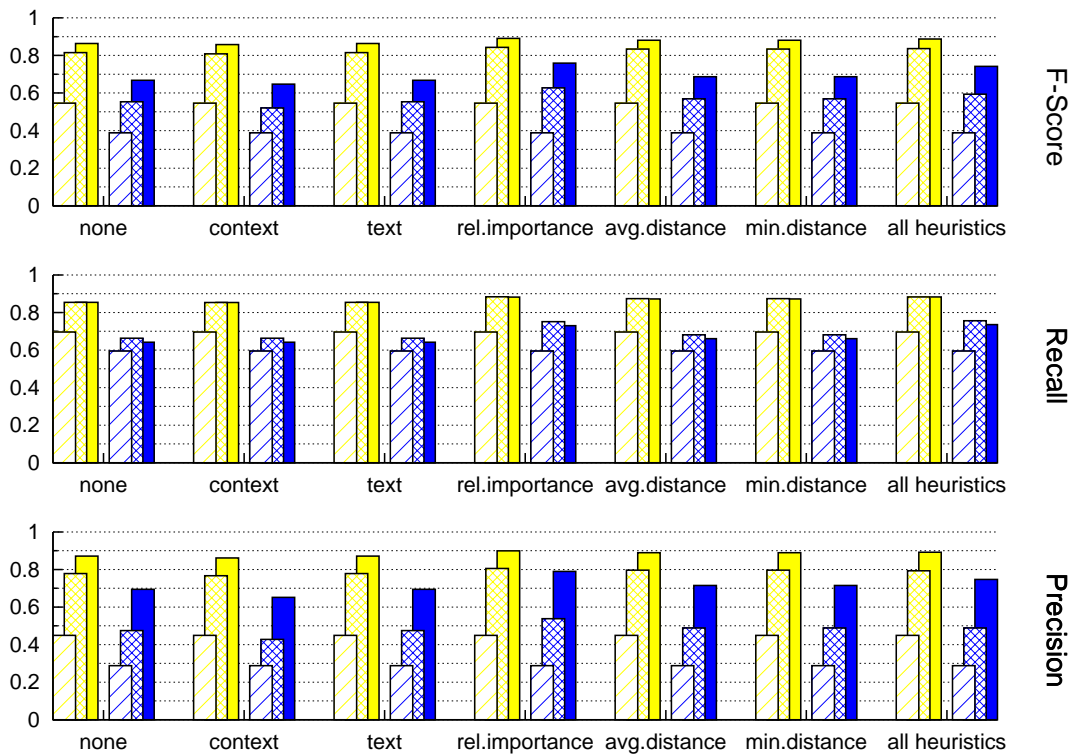
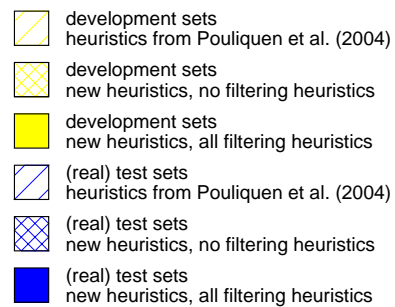


Figure 3.6: F-Score, recall and precision for the place name disambiguation heuristics with all (solid bars) and no filtering heuristics (cross-hatched bars) used. Shallow-deep threshold is 2. The single-hatched bars represent the results from the heuristics used by Pouliquen et al. (2004).



et al. (2004). Again, a combination of all heuristics has the best overall results - an increase of 91.0% to 0.74 for the test sets compared to the baseline algorithm. However, the relative importance heuristic shows similar - sometimes even superior - results, and therefore seems to be the most valuable disambiguation heuristic. On the other hand, the geo-context-based triggering leads to worse results as if no disambiguation heuristic would have been used. The heuristic comparing the locations to other place names in the text ('text') cannot improve the results either. However, when used in combination with the other heuristics, the 'text' and 'geo-context' measures positively affect the result: In some cases they are able to turn the balance to a correct place alternative, but they need a strongly weighed relative importance heuristic which reduces their influence.

**Recall.** The recall values of the single heuristics are all on a similar level, but are significantly higher than the baseline recall. The relative importance heuristic shows the highest recall, followed by the two distance measures, which have identical recall values. On the other hand, the context-based triggering and the heuristic comparing the locations to other place names in the text cannot improve the results. What is interesting is that, when the place filtering heuristics are used, the recall values tend to be lower than if they are not used. This is due to the use of the PN module, which was found to decrease the recall values in section 3.3.2.

**Precision.** Here again each single disambiguation heuristic leads to better results as the heuristics from Pouliquen et al. (2004), with the highest precision values produced by the relative importance heuristic and the distance measuring techniques. Again the 'geo-context' and 'text' measures cannot improve the results compared to the setup where no disambiguation heuristic is used.

### 3.3.4 Different languages

For measuring how well the heuristics perform for different languages, the development and test sets were analysed with all place filtering and disambiguation heuristics (solid bars in figure 3.7); the shallow-deep threshold has been set to 2. The results of the baseline algorithm from Pouliquen et al. (2004) are represented by single-hatched bars.

**Overall results.** The place filtering and disambiguation heuristics work equally well for all five languages analysed: as figure 3.7 shows, the F-Scores vary between 0.838 and 0.927 for the development sets and between 0.687 and 0.788 for the test sets. In any case, for each language the new heuristics produce significantly higher F-Scores than the baseline algorithm by Pouliquen et al. (2004).

**Recall.** For all languages, the heuristics from this work lead to higher recall values as

the ones from Pouliquen et al. (2004). However, since the results of the original algorithm by Pouliquen et al. (2004) vary rather much, the relative improvement is only minimal in some cases. However, recall values between 0.652 and 0.955 are a reasonable result when taking into account that many references in the database are not included in the gazetteer - see section 3.3.5 for more details.

**Precision.** When looking at the precision values, the benefit of the disambiguation heuristics is most apparent. Even for languages where the baseline heuristic leads to precision values of 0.40 and less, the new heuristics have precision scores of 0.80-0.95 (development sets) respectively 0.70-0.80 (test sets). This improvement is mainly due to the fact that false hits are effectively sorted out: in most cases, the number of false hits is reduced by more than 80%. In the case of the Spanish development sets, the number of false hits decreases even by 97.4% (see also table A.5 on page 73).

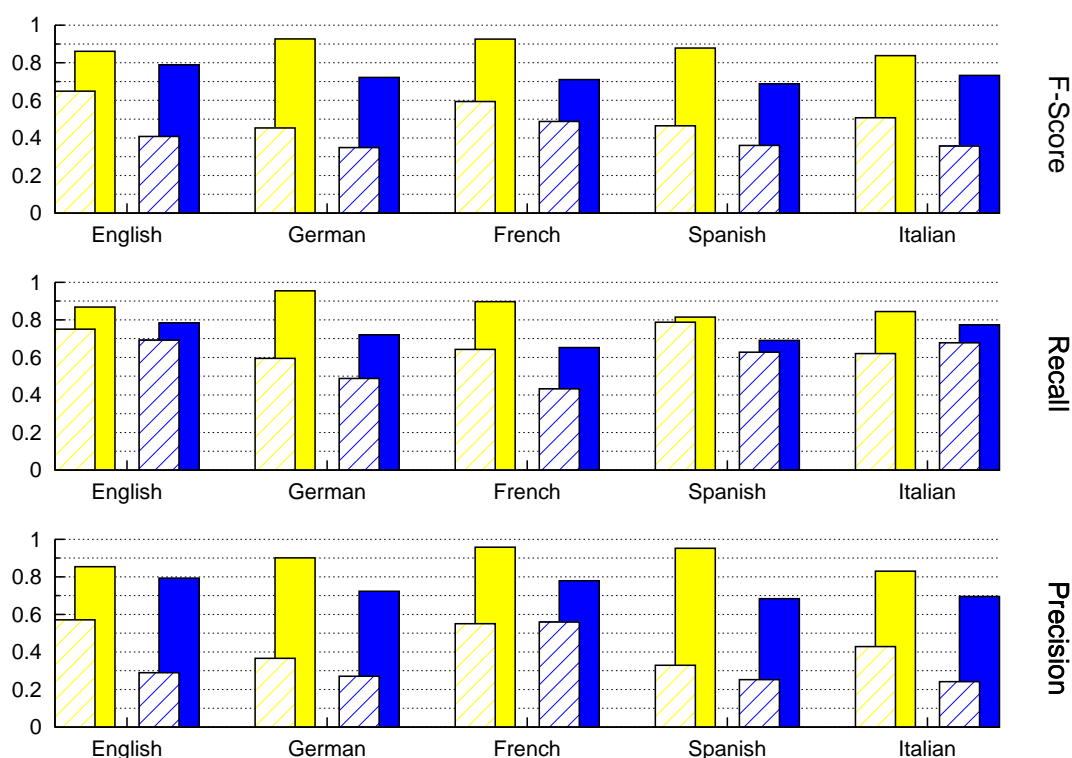
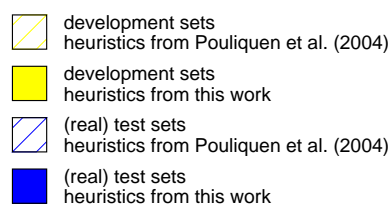


Figure 3.7: F-Score, recall and precision for the analysed languages. The whole set of filtering and disambiguation heuristics has been used, shallow-deep threshold is 2. The hatched bars represent the results from the heuristics used by Pouliquen et al. (2004).



### 3.3.5 Topics

For measuring how well the heuristics perform for the various test clusters, the development and test sets were analysed with all filtering and disambiguation heuristics (solid bars in figure 3.8), the shallow-deep threshold has again been set to a value of 2. For comparison reasons, the results of the baseline algorithm by Pouliquen et al. (2004) have been added to figure 3.8 (single-hatched bars).

**Overall results.** Although the new heuristics perform significantly better for each of the four topics, the absolute F-Scores differ pretty much: in comparison to the close-to-perfect F-Score of 0.96 for the *Turkey* cluster, the results for the *Reagan* cluster (F-Score: 0.66) seem disappointing. Due to these major differences, the following presentation will

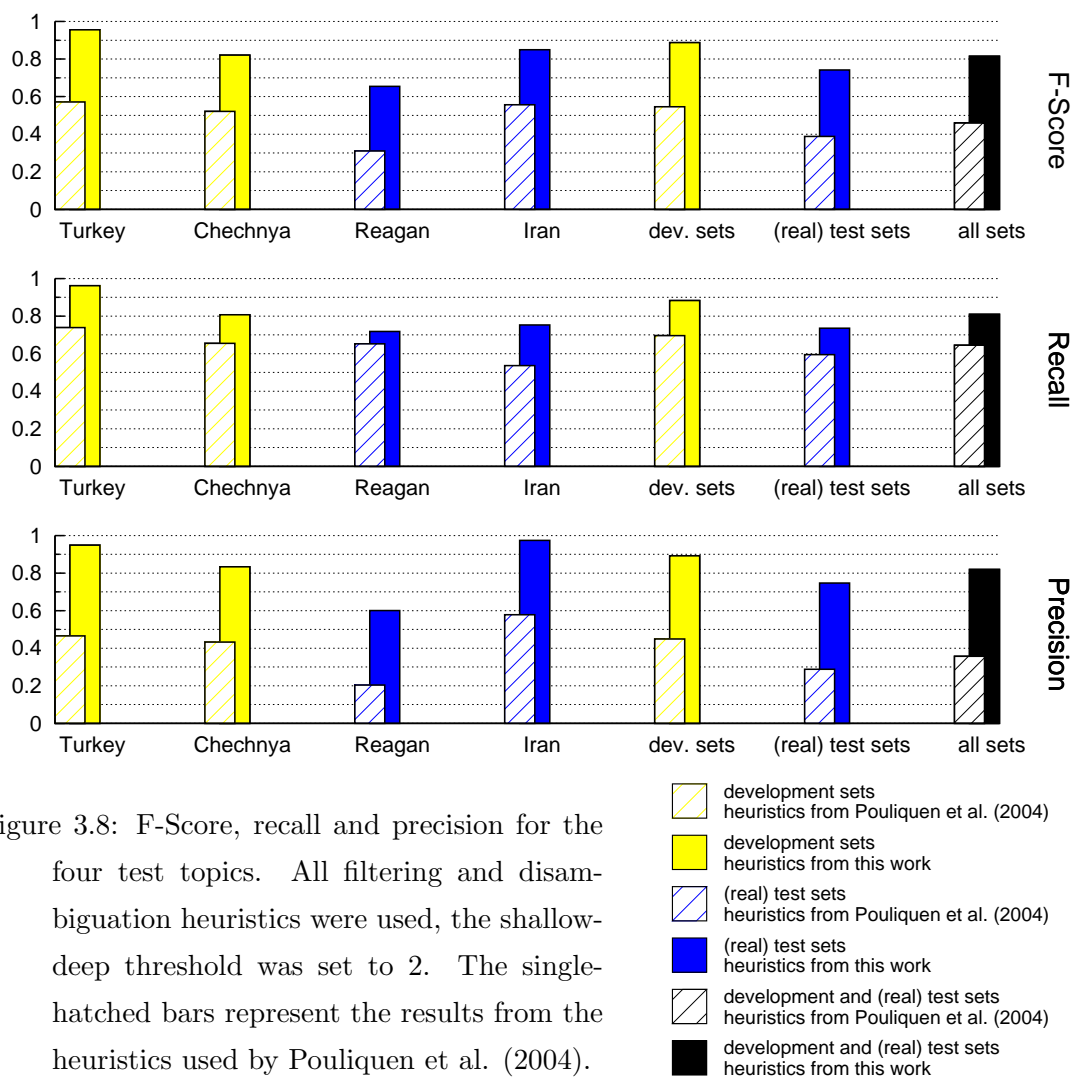


Figure 3.8: F-Score, recall and precision for the four test topics. All filtering and disambiguation heuristics were used, the shallow-deep threshold was set to 2. The single-hatched bars represent the results from the heuristics used by Pouliquen et al. (2004).

focus on the single topics and therein will look at the precision and recall values. In the previous sections, this has been done vice versa.

**Turkey cluster.** With a F-Score of 0.96 the best results were achieved for the Turkey cluster, with both the precision and recall values being at the same level. The relative improvement of the F-Score in comparison to the baseline algorithm (67.4%) is impressive. The few false hits were mostly due to media companies with a geographical reference in their name, which was wrongly estimated to refer to a place (CNN *Turk*, news agency *Anadolu*). False hits due to person names not filtered out are only a minor problem in this cluster: most of the actors named in the texts were either politicians contained in the VIP-lists, or they were filtered out by the heuristical name detector.

**Chechnya cluster.** The new heuristics' F-Score for the Chechnya cluster is 0.82, with comparable precision and recall values. In comparison to the heuristics used by Pouliquen et al. (2004), the F-Score increases by 63.1%.

Recall is affected by the fact that some alternative spellings are not contained in the gazetteer: Though the Chechen capital "Grozny" has 24 spelling alternatives in the gazetteer, the Spanish alternative "Grozni" is not contained.

Another problem affecting both recall and precision is that undeclined places are in any case given priority to declensions - in this case the disambiguation heuristics are not used. An example: many English articles contain "Chechen" or "Chechens"; but since undeclined words are given priority to declensions, "Chechen" is being related to a class-6 village situated in Russia's far east, close to Alaska, rather than to "Chechnya", of which "Chechen" is a declension.

For a few texts, no geo-context could be set because the texts do not contain references which could be detected in the first pass of the shallow-deep parsing process. For example, there are two texts containing "Grozny" as their only geographical reference; since Grozny is assigned class "6"<sup>18</sup> in the gazetteer, it was not detected in the shallow parsing step.

**Reagan cluster.** The Reagan cluster shows the worst absolute results of the four topics being analysed (F-Score: 0.66). But since also the heuristics from Pouliquen et al. (2004) lead to a poor F-Score of 0.31, a clear improvement of the results can be observed (+110.6%). For the Reagan cluster, the new heuristics even lead to precision values which are nearly three times higher than the ones achieved by the baseline algorithm - this is by far the most significant improvement which could be observed in this study.

---

<sup>18</sup>The fact that the capital of a Russian province with more than 200,000 inhabitants (Wikipedia 2004b) is assigned class "6" can be considered as an error in the gazetteer. This shows how dependent a gazetteer-based approach is on the quality of the data it contains.

The low precision value of only 0.60 has multiple reasons: First, many of the texts contain relatively many person names which are homonyms with place names; since these names are not contained in the VIP-lists, and since they have not been sorted out by the heuristical person name detector, they are counted as false hits. Second, the texts contain many references of buildings which have been wrongly assigned to other places (*Capitol*, *White House*, *Presidential Library* etc.). Third, many false hits due to place names which are also frequent words in a language (e.g. *Cold War*, *D-Day*, *Friday*) show that the geo-stop-lists in the present form are not sufficient for filtering out most of such homonyms. An automatically updated geo-stop-list, to which words are added that have mainly raised false hits in previous analyses, could certainly improve the results.

The rather low recall values of 0.719 are due to the fact that some places or place alternatives are not contained in the gazetteer. In some cases, this has even consequences for both recall and precision. An example: many texts contain the reference *Hollywood*. Although the gazetteer contains nine entries for Hollywood of which seven are in the US, the one standing for the centre of the US movie industry is not contained. The reason is that this Hollywood is just a district of Los Angeles, and no autonomous place, and districts are not contained in the gazetteer. In consequence, even if the disambiguation heuristics work perfectly, the reference cannot be detected (lowering recall), and furthermore a false hit lowers precision.

The problems many texts from the *Reagan* cluster encounter indicate that the heuristics have problems with newswire texts which do not report on a recent event (“breaking news”), but illuminate the background of an event - in this case the presidency of Ronald Reagan. Since the relative improvement of the results is even higher than for the other test topics, the bad results are not to be ascribed to the fact that the Reagan cluster was used as a test set rather than a development set.

**Iran cluster.** For the Iran cluster, the new heuristics show a close to perfect precision (0.974), but a rather bad recall of 0.579. Again, the relative improvement compared to the baseline algorithm (+68.2%) is striking.

The good precision values prove the effectiveness of the disambiguation heuristics: many of the texts include the place name *Bam*, where a severe earthquake occurred five months before the earthquake being reported on in the articles. Despite the fact that the extended gazetteer contains 12 entries for *Bam*, of which three are in Iran and two are closer to the actual epicentre of the article, the correct alternative has been assigned in nearly all articles. The low recall is mainly due to the fact that many provinces and small villages mentioned in the text are not contained in the gazetteer.

### 3.3.6 Running time analysis

For examining how long the analysis of texts with the new heuristics from this work takes, and what proportion of the total analysis time each heuristic needs, two analyses were performed: at first, 10 randomly chosen English texts were analysed, and the time needed by each heuristic was measured. Then, all development and test texts were analysed with different analysis options to get a more comprehensive impression of how long the analysis of a text takes, and how the running times are distributed.

**Proportion of each heuristic to total running time.** For measuring the proportion of time each heuristic needs in comparison to the total running time, 10 English texts were randomly chosen and analysed. In addition to the analyses done before, the running time of certain parts of the analysis was measured. In this way, the running time of each heuristic<sup>19</sup> could be estimated. Moreover, the overhead time was measured, i.e. that part of the total running time, which was not used by any heuristic. The overhead time includes e.g. the geo-parsing phase and a mapping of the analysed text into an internal representation, which allows for an easy processing.

Figure 3.9 shows the proportion each heuristic needs in comparison to the overall running time. The average running time of the whole analysis is 6.8 seconds. It is easily visible that most of the time (more than 80% or 5.5 seconds) is needed by the place triggering heuristic. Although this heuristic has not yet been optimised for running time,

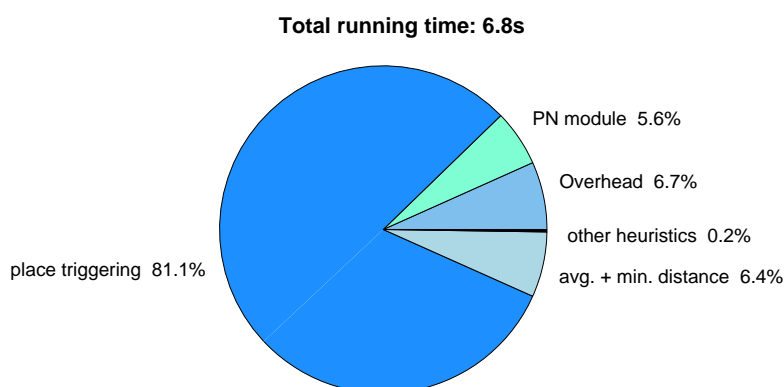


Figure 3.9: How much time does each heuristic need compared to the overall running time?

<sup>19</sup>An exception are the distance measures: both the minimal and the average distance heuristic use the same code for calculating distances, and only the relatively small part of calculating the minimal respectively the average distance differs. Therefore, there is only one time estimation for both heuristics.

it is believed that its performance cannot be boosted enough so that it would be fast enough in a production environment, especially since it has been shown in section 3.3.2 that the heuristics cannot improve the result at all.

The PN module and the two distance heuristics each take on average 0.4 seconds per text, which is the same magnitude as the overhead. This running time is acceptable since in sections 3.3.2 and 3.3.3 it has been shown that these heuristics can improve the results. The other heuristics need a negligible amount of time (each less than 0.02 seconds) - for them time will certainly not be a problem. An analysis of longer texts showed that the running time of the heuristical name detector and to a lesser extent of the distance heuristics increases faster than the other heuristics. In other words, the proportion of the time needed by the PN module and the distance heuristics increases for longer texts.

Since it is obvious that the place triggering heuristic, at least in the current version, is not worth using in a production environment, it will be excluded from further running time analysis.

**The distribution of running times.** For estimating the running time of the analysis for different texts, all 161 development and test texts were analysed with different analysis options. The place triggering heuristic was not used. In total, 6,762 analyses were performed. For each analysis, the time required was measured. The single running times were then grouped into intervals of 0.1 seconds, which are shown as blue bars in figure 3.10. The yellow line in figure 3.10 indicates what percentage of the analysed texts is analysed in a certain time (or less).

The median of all running times is 0.82 seconds, that means that 50% of all articles are analysed in less than that time. On the other hand, a few texts need up to ten seconds to be analysed. These outliers lead to a substantial skew of the data set to the right. The outliers also lead to a relatively high average running time of 1.31 seconds, although more than 70% of all texts were analysed in less than 1.3 seconds.

It should be noted that the running time analysis was performed on a system, on which the gazetteer (an Oracle 8i database) was installed on a desktop PC under Windows XP. Due to the demanding hardware requirements of an Oracle database, this is certainly not an ideal high-performance setup. A test using MySQL on a Linux server for the gazetteer lead to running times which were around 30% better than the ones achieved by the system with the local Oracle database. Therefore we think that the performance can be enhanced significantly if the gazetteer is stored in database on an adequate server.

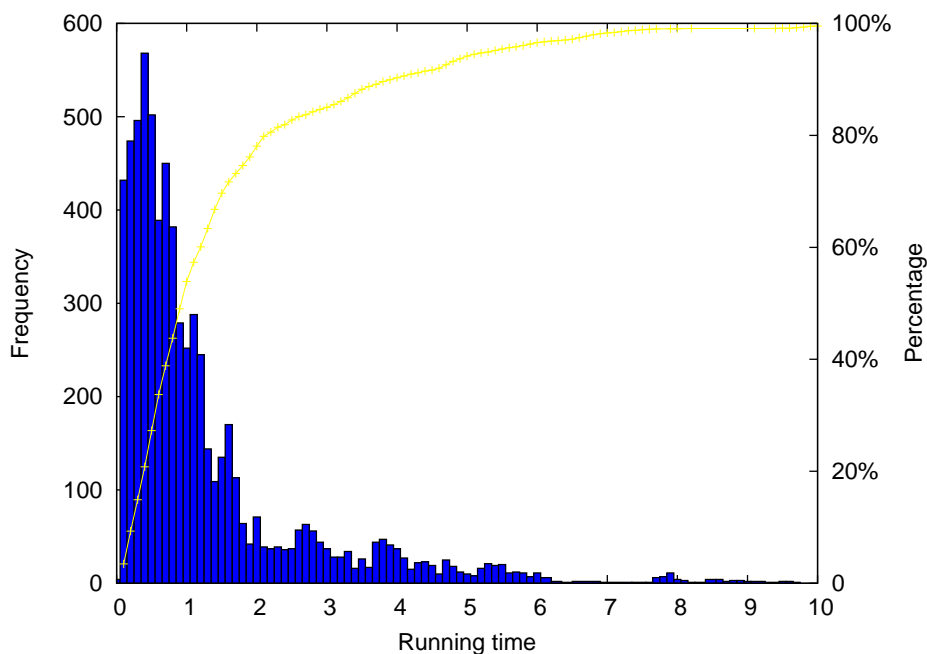


Figure 3.10: Distribution of running times for 6,762 analyses with the heuristics developed in this work.

### 3.3.7 Summary of the results

In the previous subsections, the results for the analysis of the development and test sets were analysed with the heuristics introduced in this work. In each subsection, the focus was laid to another aspect.

It was found that the newly introduced shallow-deep parsing can significantly increase the performance of the heuristics. The best overall results are observed when setting the shallow-deep threshold to a value of 2, i.e. if in the first iteration only countries and regions (class 0), capitals (class 1) and major cities (class 2) are queried.

With the exception of the place triggering heuristic, which does not improve the results and furthermore takes most of the time of the complete analysis, all heuristics help to improve the overall results. Compared to the baseline algorithm presented in Pouliquen et al. (2004), the increase in performance is significant. Especially the number of false hits is considerably lower, but also the number of missed hit declines, while more correct hits are detected. Consequently, a significant improvement of both precision and recall can be observed. The results show that a combination of the heuristics leads to the best overall

performance. An error analysis showed that the main problem is that many person names are wrongly assumed to be places, the disambiguation between place alternatives, on the other hand, is quite reliable.

The heuristics work equally well for the five languages taken into consideration. Differences in the performance between languages are not to be ascribed to the heuristics, but to the amount of entries for the various languages contained in the gazetteer.

In contrast to that, the various types of texts indeed affect the results. Texts about a certain event (as the *Turkey*, *Chechnya* and the *Iran* clusters) lead to very promising precision and recall values of more than 0.80 - texts with a close-to-perfect F-Score of more than 0.95 are quite common. On the other hand, texts providing background information about a topic lead to significantly lower F-Scores. These texts contain exceptionally many person names which are not yet reliably detected. These undetected person names are the main reason that the texts from the *Reagan* cluster have a F-Score of just 0.66.

It was found that the place triggering heuristic takes most of the analysis time - around 80% or 5.5 seconds. All other heuristics together, including the overhead time for parsing and pre-processing the text, need just 1.3 seconds on average. The distribution of the running times of 6,762 analyses is, due to a few texts which need up to 10 seconds for being analysed, strongly skewed to the right, with the average running time being 1.3 seconds and the mean running time being 0.82 seconds.

## Chapter 4

# Visualisation

In chapter 3, geographical references in texts have been recognised and assigned to their real-world counterparts. The problem of *visualising* the detected place names, however, has not been addressed yet.

Pouliquen et al. (2004) provide a HTML output of the analysed text and highlight the identified place names with colour, and additionally provide the country the place lies in (figure 4.1). This visualisation keeps the references in their original context and gives - by providing the country the reference lies in - an indication whether the right alternative has been chosen. On the other hand, this text-based visualisation does not give a spatial overview, which is provided for example by maps.

Furthermore, Pouliquen et al. (2004) as well as other authors use online map services (like MapQuest) for presenting the detected references on a map. The shortcoming of these map services is their inflexibility: They mostly produce static maps, and do not allow for user interaction. Moreover, it is not possible to display additional information

[Turkish](#)*[Türkiye/tr]* police have arrested 16 suspected al Qaida linked militants believed to have been planning a bomb attack during a Nato summit in [Istanbul](#)*[Istanbul/tr]* in June. Prime Minister Tony Blair and President George Bush are expected to attend the Nato meeting. The 16 were held in an operation in the north-western province of [Bursa](#)*[Bursa/tr]*, a police statement said. It said the suspects were members of the Ansar al-Islam, a group linked to the al Qaida terrorist network, but gave no further details. The Nato summit will mark the formal entry into the alliance of [Romania](#)*[Romania/ro]*, [Bulgaria](#)*[Bŭlgariya/bg]*, [Latvia](#)*[Latvia/lv]*, [Estonia](#)*[Estonia/ee]*, [Lithuania](#)*[Lithuania/lt]*, [Slovenia](#)*[Slovene/si]* and [Slovakia](#)*[Slovensko/sk]*.

Figure 4.1: Visualisation of detected place names in HTML, as used by Pouliquen et al. (2004).

like the context of a place name or alternative locations, which have been sorted out in the disambiguation phase.

This chapter presents a map visualisation supporting these features. Specifically, the visualisation of the detected geographical references should fulfil the following requirements:

- All “relevant” information should be included. This is, obviously, the position of the locations detected and their place names, but also additional information like the context of the occurrences, or alternatives which have been sorted out in the disambiguation phase.
- The user should be able to interactively zoom in the map, and to move that part of the map which is currently displayed.
- The user should be able to show or hide parts of the available information. It should, for example, be possible to hide place names by clicking a certain button. Some information should be visible only if the user moves the mouse over the place name it is connected to.

After an analysis of the visualisation formats presented in section 2.3, SVG was chosen for visualising the found references. It offers built-in scrolling and zooming behaviour; hence there is no need for manual implementation. SVG allows to group information, which can be hidden or shown by using ECMA scripting functions, and it supports mouse-over effects. Macromedia Flash would fulfil the requirements as well, but it has not been chosen since its proprietary binary format is expected to be more complex to process. VRML97 and X3D additionally would add a third dimension and advanced lighting models, but since these features are not needed here, the increased development does justify their use.

For this work, no special map projection was used, i.e. the earth’s surface is mapped to a rectangular 2D map, in which the grid of the earth’s coordinate system appears as a set of perpendicular straight lines. Although this simple projection does not very well follow the properties of an ideal map (realistic representation of areas, distances, angles, and shapes) (Pearson 1990), it seems sufficient for a first map representation of geographical references.

The rest of this chapter is structured as follows: The overall data model for visualisation will be described in section 4.1. After presenting an overview of the developed visualisation, section 4.2 will focus on the single features which are part of the visualisation: country shapes and background (4.2.2), place names (4.2.3), context of the detected references (4.2.4), place name alternatives (4.2.5), and the user interface (4.2.6).

## 4.1 Data model for visualisation

To separate the content from the core functionality, the data for a map visualisation is stored in two different files:

- The actual map data (country shapes, places and place labels, contextual information, alternative places etc.) is stored in a SVG file.
- The programming logic is encapsulated in a single ECMA script file called *map-Functions.js*, which is used by every SVG file. This script file contains all script functions needed to provide the desired zooming and scrolling behaviour, and processes the events from user interactions.

The following subsections will provide further information on the SVG data and the script file.

### 4.1.1 SVG data

The data to be visualised is stored in a SVG file. The map data is grouped in eight layers, of which the first five are displayed in figure 4.2:

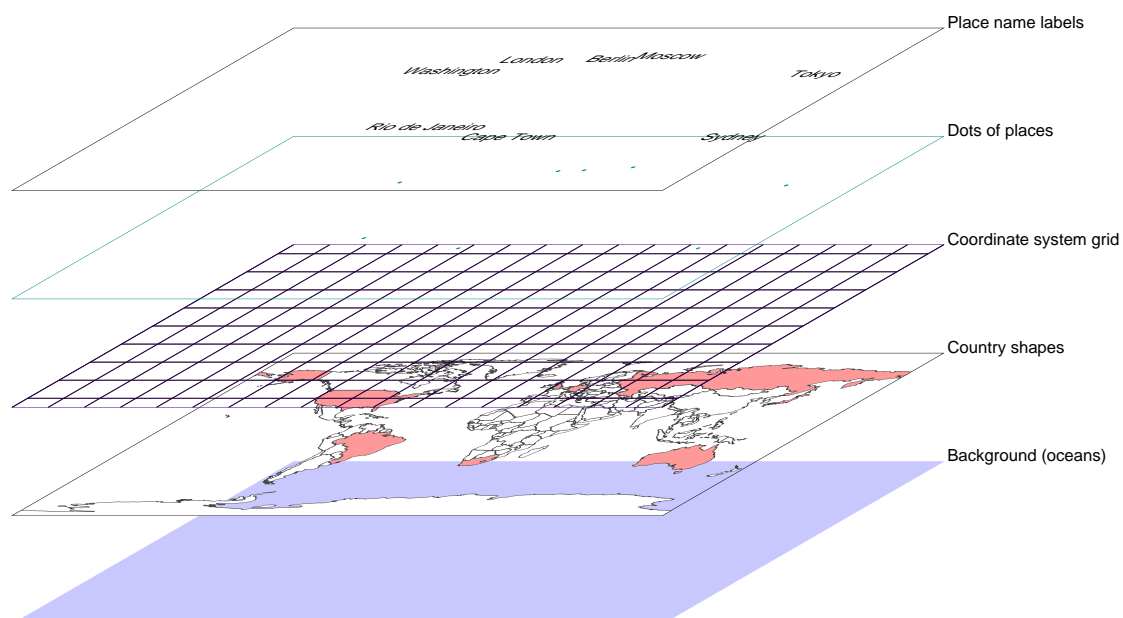


Figure 4.2: Visualisation of five SVG layers.

1. The *background* layer just contains a light blue rectangle, which is only visible if it is not covered by another layer. It represents oceans and seas.
2. The *countries* layer contains the shapes of all countries in the world. This layer is unique to all generated maps, apart from the fact that countries comprising place references are coloured differently.
3. The *grid* layer contains a grid representing the earth's coordinate system.
4. The *dots* layer contains small circles, representing the position of all the places detected in the text analysed.
5. The *places* layer contains the place name labels of all places detected.
6. The *context* layer contains all occurrences of the places detected, and the respective context in the text.
7. The *alternatives* layer contains the places which have the same name as a detected place reference. These alternative place names were sorted out in the disambiguation phase.
8. The *infobox* layer contains all elements making up the user interface.

Figure 4.3 shows the corresponding SVG code for these eight layers. Details for each group are given in section 4.2.

### 4.1.2 Programming logic

While the SVG file contains the actual map data, the functions processing the user input are encapsulated in a single ECMA script file called *mapFunctions.js*. The functions are called when an event is triggered in the SVG map.

An example clarifies the event-handling procedure used by SVG: The following code is a fragment from a SVG file which has included *mapFunctions.js* as its script file:

```
<text onclick="showInfoForPlace( foo )">  
    More information on place foo  
</text>
```

If the user now clicks the mouse button while the mouse cursor is over the text, the function `showInfoForPlace()` is called with the parameter `foo`, which then shows - as the function name suggests - additional information on *foo*.

In our case, *mapFunctions.js* ...

- ... corrects size and position of elements which should keep their position on the screen, even if the map is scrolled or zoomed,
- ... manages user preferences, like the font size which should be used for displaying text,
- ... displays or hides the layers the user has chosen to see (or not to see),
- ... calculates and displays the latitude and longitude values the mouse cursor currently points on.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000303 Stylable//EN"
  "http://www.w3.org/TR/2000/03/WD-SVG-20000303/DTD/
  svg-20000303-stylable.dtd">
<svg viewBox="-180 -90 360 220" width="100%" height="100%"
  onload="setup(evt)" onscroll="resetInfo(evt)"
  onresize="resetInfo(evt)" onzoom="resetInfo(evt)"
  "preserveAspectRatio="xMidYMid meet">
  <script xlink:href="mapFunctions.js" type="text/ecmascript" />
  <g id="background" onmousemove="findUserCoord(evt);"> [...] </g>
  <g id="countries" stroke-width="0.10" stroke-miterlimit="4"
    onmousemove="findUserCoord(evt);"> [...] </g>
  <g id="grid"> [...] </g>
  <g id="dots"> [...] </g>
  <g id="places"> [...] </g>
  <g id="context"> [...] </g>
  <g id="alternatives"> [...] </g>
  <g id="infobox"> [...] </g>
</svg>
```

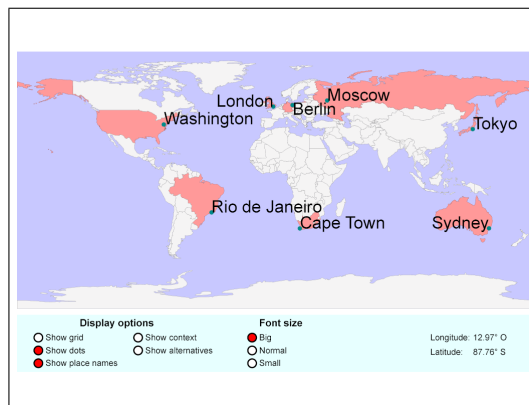
Figure 4.3: Structure of a SVG map file. Each group tag `<g>` stands for one layer, `[...]` is a placeholder for the elements the group/layer contains.

## 4.2 Visualised features

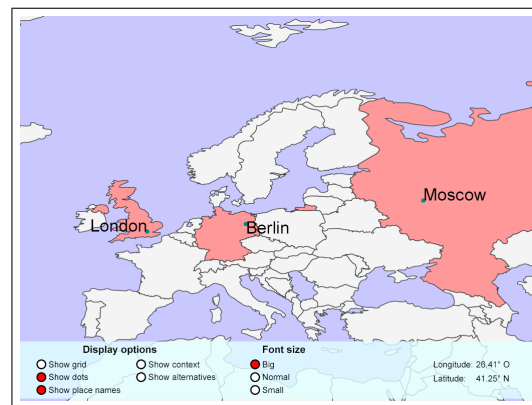
### 4.2.1 Overview

Figure 4.4 shows some screenshots of the SVG map visualisation, which highlight the most important features:

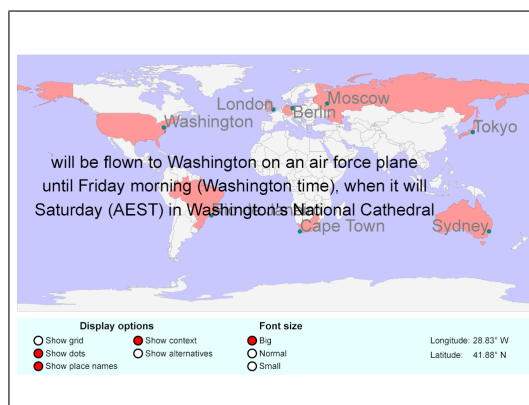
- Figure 4.4a shows the original view of a map with major cities, which is displayed right after the SVG has been retrieved and rendered by the SVG viewer. The places which were detected by the analysis of the text the map was created from are displayed, and the countries containing references are coloured.



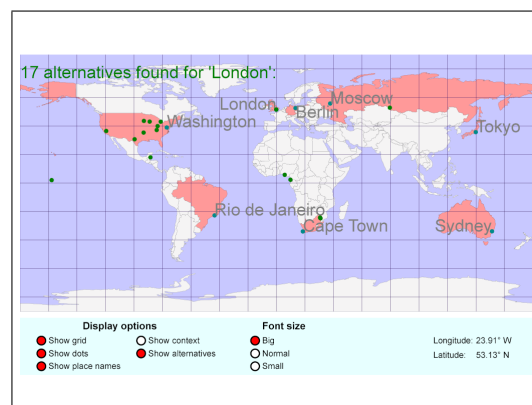
(a) original view



(b) zoomed in



(c) context for reference "Washington"



(d) alternative places for reference "London"  
with the coordinate system grid being enabled

Figure 4.4: Screenshots of a SVG map highlighting the most important features of the visualisation.

- The map is fully zoomable and scrollable. In figure 4.4b the map has been zoomed to show only Europe. The size of the place dots and place labels remains constant as well as the size and position of the user interface.
- The user can simply show or hide additional information like the earth's coordinate system (figure 4.4d). Additionally, when moving the mouse over a place label, the user can get additional information on the context of a reference in the original text (figure 4.4c), or on places sharing the name with a reference (figure 4.4d).

Further details on the implementation of the features will be given in the following.

### 4.2.2 Country shapes and background

From an implementation point of view, the country shapes and the background are straightforward to realise: they never change their position relative to the world coordinate system, and they should zoom regularly. Since scrolling and zooming are part of the built-in functionality of SVG, the programmer does not have to implement them manually. Furthermore, the country shapes are displayed for all maps which are generated, so no individual adaptation is necessary.

As a basis for the country shapes used in this work, an ESRI shapefile contained in the Global Discovery Gazetteer was used. The coordinates of the country shapes, which were stored as positive values in an ordinary coordinate system, were converted into latitudes and longitudes in the earth's coordinate system. Additionally, all  $y$ -values were inverted, because in SVG coordinate systems latitude values on the northern hemisphere are represented by negative  $y$ -values. The country shapes were stored as a SVG fragment, which is included when a new SVG map is generated.

In contrast to the country shape, the filling colour varies according to the number of geographical references detected and displayed in a country: countries for which no references are contained in the text are filled with a light grey background; countries with references are filled in red, with the red becoming more intensive when the country contains more references. Table 4.1 shows the colour shades with which the country shapes are filled.

References	Colour
0	Light grey
1 – 2	Light red
3 – 5	Medium red
6 – 10	Dark red
> 10	Very dark red

Table 4.1: Background colours of countries

### 4.2.3 Detected place names

At first glance, also the visualisation of detected references seems to be simple: Each place has a coordinate (consisting of latitude and longitude), and at this point a dot can be drawn onto the map. The coordinate could also be used as a reference point for the label of the place (i.e. the place name): by default, in SVG this reference point is the lower left corner of the rectangle enclosing the text, i.e. the text is printed on the top right side of the point.

However, there are two problems with this straightforward solution: At first, the relative position of the places is not taken into account; hence, since all place labels are printed on the top right of the place's coordinate, labels of vicinal places may overlap. See figure 4.7a for an example of overlapping place name labels. In order to prevent such overlaps, a label-alignment algorithm is used, which will be described further down in this section.

The second problem is that, due to the default zooming behaviour, the place dots and the place name labels are also zoomed. This leads to text in huge font sizes, as shown in figure 4.5b. To prevent this undesired zooming behaviour, the font size is adjusted when the zoom factor changes, so that the font size remains constant (figure 4.5c). The position of the place dot and the anchor point of the place name label, however, does not change.

#### Label alignment algorithm

Assumed that the coordinates of a point  $P$  are in the origin of a coordinate system, the label of this point lies in one or two adjacent quadrants.<sup>1</sup> To keep the problem simple,

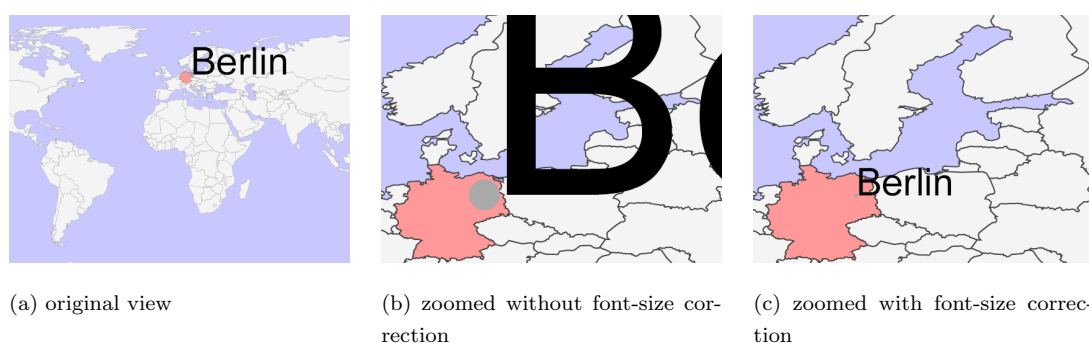


Figure 4.5: Example of SVG zooming behaviour

---

<sup>1</sup>Theoretically, the label could also lie in all four quadrants. Then, however, the label would conceal the dot, which is usually not desired in maps.

we restrict to labels lying in just one of the four quadrants, i.e. one of the corners of the rectangle enclosing the label touches the point's coordinate.

But which of the quadrants should the label lie in so that the probability that labels overlap is minimal? Simply speaking, the key to the problem is to avoid regions containing many places to be displayed.

To detect such regions, a weighed central point  $C$  for a place  $P$  is calculated. This weighed central point is comparable to the centre of gravity of masses, whose coordinates  $x$  and  $y$  are defined by

$$x = \frac{\sum m_i x_i}{\sum m_i}, \quad y = \frac{\sum m_i y_i}{\sum m_i} \quad (4.1)$$

where  $m_i$  ( $i = 1, 2, \dots, n$ ) is the mass of a material point  $i$ , and  $x_i$  and  $y_i$  are its coordinates (Bronstein et al. 1999, eq. 3.273). In our case,  $m_i$  is defined as  $\frac{1}{d_i}$ , where  $d_i$  is the distance of place  $P_i$  to place  $P$  in units of degrees.<sup>2</sup> Figure 4.6 shows an example of such a central point  $C$  for a point  $P$ .

If again point  $P$  is the origin of a coordinate system, the optimal quadrant for the place name label is the one which is opposite to the central point  $C$ . In the example in figure 4.6 this is quadrant  $IV$ , i.e. the place name label would be positioned on the lower right side of point  $P$ .

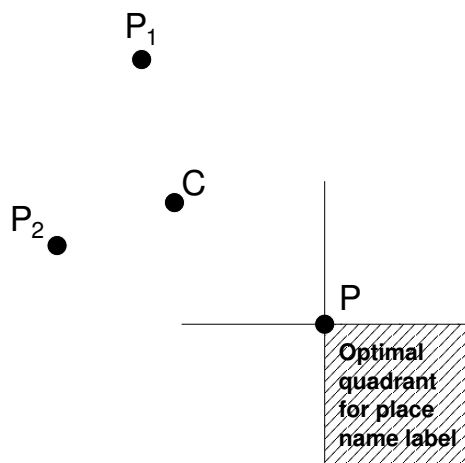


Figure 4.6: If  $C$  is the weighed central point of  $P$ ,  $P_1$  and  $P_2$ , the label for place  $P$  should be placed in the hatched quadrant.



(a) default label alignment



(b) Labels aligned with algorithm

Figure 4.7: Labelling place names.

<sup>2</sup>To avoid extreme values, the maximum of  $m_i$  is set to 1, and places  $P_i$  with  $d_i > 30$  are discarded ( $m_i = 0$ ).

#### 4.2.4 Context of the geographical references

As indicated in section 4.1.1, the context of the detected references is stored in its own SVG layer. This layer contains a group of `<metadata>` tags for each detected reference. The words surrounding each of the occurrences of a reference (which is defined as the context) are stored in `<context>` tags inside the `<metadata>` tag. Figure 4.8 shows an example of the context layer and its visualisation in a web browser.

The context of a reference is only shown when the mouse cursor is moved over the place label (onmouseover-event), and if the option “Show context” is enabled in the user interface. The context - i.e. the surroundings of the reference - is displayed below the mouse cursor. For better contrast, the colour of the place name labels fades to grey while the context is displayed (see figure 4.8 for an example).

```
<g id="context">
  <g id="context_825">
    <metadata>
      <context value="... He travelled to Berlin in 1987 to ..." />
      <context value="... to demand the Berlin Wall be torn ..." />
    </metadata>
  </g>
</g>
```



Figure 4.8: SVG fragment and its visualisation of the context of the detected reference “Berlin”.

### 4.2.5 Place alternatives

Place alternatives are locations which have the same name as a detected reference, but have been sorted out in the disambiguation process. If the user has enabled the “Show alternatives” option in the user interface and moves the mouse over a place label, the location’s homonyms, which are represented by green dots, pop up. Additionally the number of references is displayed (see also figure 4.9).

In the SVG file, the alternatives are stored in the alternatives layer. For each reference, a group containing the information string and a `<circle>` tag for each alternative is generated. Figure 4.9 shows an example of the alternatives layer as SVG fragment.

```
<g id="alternatives">
  <g id="alternatives_825">
    <text id="alternativeLabel" x="-178" y="-80" style="fill:green">
      14 alternatives found for 'Berlin':</text>
    <circle id="altDot_189306" cx="-74.937872" cy="-39.794035" r="2"
      style="fill:green;stroke-width:0;" />
    [...]
  </g>
</g>
```

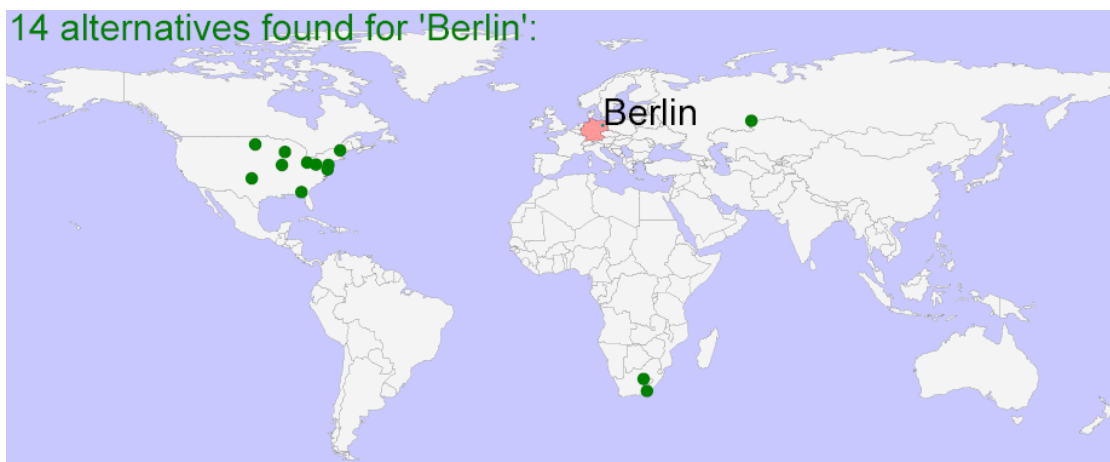


Figure 4.9: SVG fragment and its visualisation of the alternatives of the detected reference “Berlin”.

### 4.2.6 User interface

With the user interface the user can easily change display options. Furthermore, additional information is shown in the user interface panel.

On the left side of the user interface panel, several display options can be enabled or disabled. By clicking on the check boxes left of the labels, the user can toggle the display of the feature. If a check box is filled red, its respective feature is displayed in the map.

In the middle of the panel, the user can change the font size of the place name labels and the context of the references in the map. “Normal” font size is the recommended value for display on a computer screen, “Big” results in readable texts on a printed version of the map. If the visualised text contains many references, the “Small” font size option helps to avoid overlapping place name labels.

On the right side of the panel, the current coordinates of the mouse cursor in the earth’s coordinate system are displayed. The latitude and longitude values are shown in decimal degrees, and are updated whenever the position of the mouse cursor changes.

An inevitable prerequisite for an user interface is that it is static, i.e. that neither its position nor its size change - otherwise the usability would suffer. Since by default all elements in a SVG object (including a SVG-based user interface) are zoomed and moved, this behaviour was added manually: when the zoom factor changes, the size of the whole user interface panel relative to the coordinate system is adjusted so that the size on the screen remains constant. Similarly, the position of the panel is adjusted when the map is moved. Consequently, the user interface panel keeps its position at the bottom of the map, and the size on the screen remains constant.



Figure 4.10: User interface for setting options.

## Chapter 5

# Discussion

In chapter 3, a set of place filtering and disambiguation heuristics has been introduced, which then were applied to 161 test articles in five languages. The results of this analysis have been presented in section 3.3. This chapter combines the single results of the heuristics and assesses the overall performance. Finally, the visualisation of the detected references is discussed.

The first and probably most important step in the analysis is the shallow-deep parsing: in a first iteration, only the most important places in the gazetteer are queried; out of the results of this shallow parsing a geographical context is inferred. A second step then queries the gazetteer in its whole depth, but restricts to places in those countries which were found to account for the geo-context of the text. This two-step approach successfully limits the amount of data being queried, and therefore obviates many potential false hits. On the other hand, a second iteration - called deep parsing - assures that all entries in the geo-context of a text are returned, so that no (or only few) correct hits are missed. The tests presented in section 3.3.1 show that the best results are achieved for a shallow-deep threshold of 2, which filters out between 50% and 80% of the false positives produced by the heuristics from Pouliquen et al. (2004). The shallow-deep parsing works reliably for each language examined, and significantly improves the results for each of the development and test topics analysed.

The high performance of the shallow-deep parsing outshines the other heuristics, which usually improve the results in a magnitude of a few per cent. Anyway, most of the heuristics being analysed could improve the results. The only exception is the place-triggering heuristic, which had no effect at all and moreover takes most of the time for the whole analysis. Therefore, at least in the current version, this heuristic cannot be

recommended for use in a production environment. The remaining three place filtering heuristics (PN module, VIP lists, and geo-stop-lists) were found capable of filtering out many false hits, while the time they need is acceptable. However, the error analysis revealed that most false hits were due to unfiltered person names or common words of a language. Therefore, an improvement of these heuristics - which could be done automatically - could still enhance the results. See the future work section (6.2) for some ideas on how the heuristics could be improved.

For the place disambiguation heuristics, the relative importance heuristic and the kilometeric distance measures performed best. When used alone, the heuristics taking into account the geo-context and looking at other words in the text could not improve the results. When used together with the other heuristics (especially with the relative importance heuristic), the overall results improve slightly. Since it is the combination of all heuristics which leads to the best overall results, and because none of these heuristics takes too much time, it seems preferable to use all the disambiguation heuristics for analysis.

The performance differences between the languages are rather small and can only be explained when looking at the texts themselves. This in-depth analysis often reveals that the problem does not lie in the heuristics, but in an incomplete gazetteer (missing spelling alternatives for Spanish and French) or false entries in it (Why is Grozny, the capital of Chechnya, assigned class '6' = small village?). Sometimes, missing references to countries in a text make it hard to choose an alternative - even for a human reader. Therefore it can be concluded that the reasons for differences in the heuristics' performance for various languages are not due to the approach itself. Since, in addition, support for other languages which have not yet been tested can be added with little additional effort, the heuristical approach to geo-coding presented in this work can be called multi-lingual.

The performance differences for the various test clusters show how much the type and content of an analysed text can affect the results. For the most common type of newswire texts - report on a recent event - the performance is consistently good, and a significant difference between development sets and test sets cannot be observed. The heuristics have more problems with background stories, which often contain a multitude of person names or place names with no further geographical hints. This especially affects the performance of the *Reagan* cluster. But since both the baseline algorithm by Pouliquen et al. (2004) and the new heuristics from this work perform badly, the worse results are not to be ascribed to the fact that the *Reagan* cluster is a test set rather than a development set. In contrast, the relative improvement is better for this cluster than for all the development sets.

In any case, for all development and test sets the results achieved by the heuristics from this work by far outperform the ones from Pouliquen et al. (2004), and it is believed that advanced filtering techniques can further enhance the results (see above).

When excluding the place triggering heuristic, the running time of the heuristics is acceptable even for use in a batch environment. With the test setup used in this study, the average running time of the analysis is 1.3 seconds, with a median of 0.82 seconds. However, we think that the performance can be significantly improved if the gazetteer is installed on an adequate server - for the analysis presented here the gazetteer is stored in an Oracle database on a normal Windows XP desktop PC, which is certainly a bottleneck for the whole analysis. Moreover, an optimisation of the heuristics for running time might further improve the performance.

In contrast to the evaluation of the place filtering and disambiguation heuristics with F-Score, recall, and precision, it is not possible to quantify the profit of the visualisation. However, the presented SVG visualisation fulfils all the prerequisites formulated in the objectives in 1.1: The user can scroll and zoom the map, and it is possible to show or hide parts of the information - therefore the visualisation is clearly interactive. The visualisation offers previously unseen features like the display of the context of a detected reference and the possibility to show alternative places sharing the name with a detected reference. The SVG map can be automatically generated after the text has been analysed, and a link to it can be placed on any website. Hence, the maps are easy to integrate into web pages in which geographical references of a text should be displayed.

## Chapter 6

# Conclusion

The primary aim of this thesis was to recognise geographical references in texts, and to relate these references to actual place names. The geo-coding approach chosen is based on the one presented in Pouliquen et al. (2004), which restricts the usage of linguistic analysis to a minimum. The results achieved by the new heuristics are promising: For three out of the four test topics analysed the F-Score is above 0.80, with the precision being above 0.90 in many of the analysed texts. The heuristics developed in this work perform significantly better than those presented in Pouliquen et al. (2004). It can be concluded that the aim to recognise geographical references, and to relate them to actual locations is fulfilled. The quality of the results could be enhanced considerably.

One of the advantages of this approach is that it is largely language-independent. While other approaches rely on (language-dependent) NLP-approaches, this work makes little use of linguistic analyses. For adding support for a new language, only a few lists have to be created, and this generation could be performed automatically. In this work it was found that on all five languages analysed the performance (i.e. precision and recall) of the new heuristics is comparable.

The second aim of this thesis was to visualise the results on a map. This aim was fulfilled by automatically generating an interactive SVG map after the text has been analysed. The user can easily scroll and zoom the map, and has the possibility to view additional information like the context in which a place reference appeared, or locations having the same name as a detected reference. This new and tailor-made functionality, together with an enhanced usability, constitutes a clear added value to other visualisations of geographical references.

## 6.1 Contributions

The use of a gazetteer-based heuristical approach for detecting and geo-coding references is not a new idea - see section 2.4 for an overview of related work. What is new, however, is the combination of a multitude of different heuristics. In this work, four place filtering and five place disambiguation heuristics were implemented and tested. Since each heuristic focuses on one aspect of filtering or disambiguation, the combination of different types of heuristics was found to be the precondition for the superior results of this work.

While some of the heuristics have already been used and evaluated in previous work, others have only been suggested, or are completely new: The kilometeric distance heuristic has been put forward in Pouliquen et al. (2004), but was first implemented in this work; the use of spherical geometry for calculating exact distances on earth is - as far as we know - by now unique in the field of NER and geo-coding. None of the previous work done so far has tried to infer a geo-context and to use it for disambiguating geographical references. Heuristical and list-based person name detection is a new and simple way to filter person names - most works before utilised complex and mostly language-dependent NER techniques for that.

The idea to query the gazetteer in two steps has been used first in this work. This shallow-deep parsing extracts the geo-context of a text by looking at the most important geographical references, and then queries also the less important places, which lie in the countries found to be in the geo-context. This two-step approach has been found capable of limiting the amount of data being queried (and hence obviating potential false hits) while ensuring that no correct hits are missed, or only a few. Consequently, the shallow-deep parsing improves the overall results considerably.

Most of the previous work on geo-coding using visualisation techniques rely on out-of-the-box solutions (e.g. web services like MapQuest) or simple static, non-interactive visualisations. For this thesis, a dynamic and interactive visualisation based on Scalable Vector Graphics (SVG) was implemented. After the geo-coding process of a text, a SVG map visualising the detected references is automatically generated. The resulting map offers the SVG-innate features of zooming and scrolling, and has an easy-to-use interface which allows to show or hide information. Furthermore it provides hitherto unseen features as the optional display of the context of a reference and alternative places.

## 6.2 Future work

As the analysis of the results showed, the main problem of the approach presented is that many person names and words in natural language, which are found to be potential places, are not filtered out. Hence the key to optimising the results is to improve the place filtering heuristics.

Bruno Pouliquen is continuously developing his heuristical person name detector and lately has improved its performance significantly (personal conversation, August 2, 2004). Since this module has just been used as a black-box module, an enhanced version can simply be integrated and could largely improve the results, especially for texts including many person names (as, for example, the *Reagan* cluster).

Though it has been shown that the list-based heuristics work, and that they can improve the results, they are far from perfect. Although a manual enlargement and maintenance is not feasible, the lists could be enhanced automatically: the VIP lists could be updated by adjusting them with the “Persons of the day” list generated by the *Top stories* application described in Pouliquen, Steinberger and Ignat (2004). The geo-stop-lists could be maintained by automatically adding words which have raised mainly false hits in the previous analyses. A first version of such an algorithm has been already used by Pouliquen et al. (2004) for creating the geo-stop-lists; a new one could now comprise the extended gazetteer and the texts analysed meanwhile. The accuracy of the geo-stop-list heuristic could be improved by adding a probability to the list (“Place *X* has shown to raise false hits in *Y*% of all its occurrences.”), which could then influence the weighing.

Another way to improve the place filtering would be to introduce NER techniques. For this work, NER techniques have intentionally not been taken into account because they are language-dependent. However, as an additional heuristic among the others presented in this thesis, they could help to filter out named entities which are no places in a certain context. Languages for which such techniques are not used still can rely on the heuristics which have been proven to work in this thesis.

Currently, the geo-context extraction focuses on countries. In a next step, it could be extended to also take into account administrative units. This might lead to better disambiguation results for places which have multiple alternatives in one country.

By now, administrative units are not supported in the visualisation; in future development, the shapes and names of administrative units could also be shown on the map. This, however, presumes that the gazetteer includes a large number of administrative units and reliably maps places to them; in the gazetteer used so far, only a few administrative units are stored. It would also be possible to visualise more than one text in one map; this

would for example allow for an overview of regions on which the news coverage mainly focused on in a certain period of time. In the next step, changes in the news coverage could be visualised in an animation - SVG would support that.

In the current SVG map, no special map projection is used. Hence, regions near the poles (as the Antarctic, Canada, and Greenland) appear larger than regions near the equator. More sophisticated map projections would approximate the earth's surface better.

### 6.3 Outlook

Because of the good performance of the new heuristics, the presented approach will be used in a production environment at the JRC.

Initially, the heuristics will be used by the JRC's *Europe Media Monitor* (EMM) to analyse around 15,000 articles per day. More specifically, a XML representation of each article will be geo-coded, i.e. information about the detected references will be added to the XML files. The resulting geo-coded XML files will then be used by the *Top stories* application to cluster articles according to the topic they report on, and to link stories across different languages. In that context, the visualisation will at first be used for debugging purposes: on a map it is much easier for the developers to detect false hits than in the coloured text used until now. In the long term, it is planned to use the SVG maps for visualising a whole cluster of articles in the context of the *Top stories* application.

The JRC's *ISFEREA* group will include the heuristics and the visualisation on their DMA website. Via an interface it will be possible to enter a text, analyse it with the heuristics from this work, and view the results in an automatically generated SVG map; the SVG visualisation presented in this thesis will be used. In the long term, *ISFEREA* plans to integrate the geo-coding approach into their existing systems. In particular, it shall be used in the Global Disaster Alert System currently under development, where it would allow the triggering of geographic coordinate-based models based on textual information.

# Appendix A

## Results in tables

This appendix contains tables with the raw data (number of correct, false, and missed hits) and the deduced performance measures (Precision, Recall, and F-Score) from the analysis of the development and test texts with both the baseline heuristics from Pouliquen et al. (2004) and the new heuristics presented in this work. In section 3.3, the data of one table is visualised in a bar chart diagram. Table A.1 gives an overview of which tables in the appendix are connected to which figures in section 3.3.

<b>Performance of ...</b>	<b>Table in appendix</b>	<b>Figure in section 3.3</b>
Shallow-deep thresholds	Table A.2	Figure 3.4, page 36
Place filtering heuristics	Table A.3	Figure 3.5, page 38
Place disambiguation heuristics	Table A.4	Figure 3.6, page 40
Languages	Table A.5	Figure 3.7, page 42
Topics	Table A.6	Figure 3.8, page 43

Table A.1: Different aspects of the analysis with the place name filtering and disambiguation heuristics with cross references to the result tables containing the raw data and the figures visualising the results.

	<b>C</b>	<b>F</b>	<b>M</b>	<b>Precision</b>	<b>Recall</b>	<b>F-score</b>
<b>development sets, no filtering and disambiguation heuristics</b>						
Pouliquen et al. (2004)	637	781	278	0.449	0.696	0.546
threshold 1	708	217	207	0.765	0.774	0.770
threshold 2	780	195	135	0.800	0.852	0.825
threshold 3	779	293	136	0.727	0.851	0.784
threshold 4	778	418	137	0.651	0.850	0.737
threshold 5	767	645	148	0.543	0.838	0.659
threshold 6	775	835	140	0.481	0.847	0.614
<b>development sets, all filtering and disambiguation heuristics</b>						
Pouliquen et al. (2004)	637	781	278	0.449	0.696	0.546
threshold 1	736	98	179	0.882	0.804	0.842
threshold 2	808	98	107	0.892	0.883	0.887
threshold 3	806	132	109	0.859	0.881	0.870
threshold 4	803	227	112	0.780	0.878	0.826
threshold 5	802	451	113	0.640	0.877	0.740
threshold 6	807	549	108	0.595	0.882	0.711
<b>test sets, no filtering and disambiguation heuristics</b>						
Pouliquen et al. (2004)	540	1336	368	0.288	0.595	0.388
threshold 1	589	623	319	0.486	0.649	0.556
threshold 2	602	665	306	0.475	0.663	0.554
threshold 3	602	770	306	0.439	0.663	0.528
threshold 4	600	1085	308	0.356	0.661	0.463
threshold 5	597	1322	311	0.311	0.657	0.422
threshold 6	594	1501	314	0.284	0.654	0.396
<b>test sets, all filtering and disambiguation heuristics</b>						
Pouliquen et al. (2004)	540	1336	368	0.288	0.595	0.388
threshold 1	655	210	253	0.757	0.721	0.739
threshold 2	668	226	240	0.747	0.736	0.741
threshold 3	666	282	242	0.703	0.733	0.718
threshold 4	667	396	241	0.627	0.735	0.677
threshold 5	663	610	245	0.521	0.730	0.608
threshold 6	663	720	245	0.479	0.730	0.579

Table A.2: The effect of different shallow-deep thresholds.

	C	F	M	Precision	Recall	F-score
<b>development sets, no disambiguation heuristics</b>						
Pouliquen et al. (2004)	637	781	278	0.449	0.696	0.546
none	782	222	133	0.779	0.855	0.815
PN module	782	163	133	0.828	0.855	0.841
VIP lists	782	183	133	0.810	0.855	0.832
geo-stop-lists	782	162	133	0.828	0.855	0.841
place triggering	782	222	133	0.779	0.855	0.815
all heuristics	782	115	133	0.872	0.855	0.863
<b>development sets, all disambiguation heuristics</b>						
Pouliquen et al. (2004)	637	781	278	0.449	0.696	0.546
none	808	210	107	0.794	0.883	0.836
PN module	808	144	107	0.849	0.883	0.866
VIP lists	808	164	107	0.831	0.883	0.856
geo-stop-lists	808	150	107	0.843	0.883	0.863
place triggering	808	210	107	0.794	0.883	0.836
all heuristics	808	98	107	0.892	0.883	0.887
<b>test sets, no disambiguation heuristics</b>						
Pouliquen et al. (2004)	540	1336	368	0.288	0.595	0.388
none	602	665	306	0.475	0.663	0.554
PN module	583	424	325	0.579	0.642	0.609
VIP lists	602	364	306	0.623	0.663	0.642
geo-stop-lists	602	604	306	0.499	0.663	0.570
place triggering	602	665	306	0.475	0.663	0.554
all heuristics	583	256	325	0.695	0.642	0.667
<b>test sets, all disambiguation heuristics</b>						
Pouliquen et al. (2004)	540	1336	368	0.288	0.595	0.388
none	687	719	221	0.489	0.757	0.594
PN module	668	426	240	0.611	0.736	0.667
VIP lists	687	344	221	0.666	0.757	0.709
geo-stop-lists	687	652	221	0.513	0.757	0.611
place triggering	687	719	221	0.489	0.757	0.594
all heuristics	668	226	240	0.747	0.736	0.741

Table A.3: Performance of the filtering heuristics for all languages with shallow-deep threshold 2.

	C	F	M	Precision	Recall	F-score
<b>development sets, no filtering heuristics</b>						
Pouliquen et al. (2004)	637	781	278	0.449	0.696	0.546
none	782	222	133	0.779	0.855	0.815
geo-context	781	237	134	0.767	0.854	0.808
text	782	222	133	0.779	0.855	0.815
rel. importance	809	195	106	0.806	0.884	0.843
avg. distance	800	204	115	0.797	0.874	0.834
min. distance	800	204	115	0.797	0.874	0.834
all heuristics	808	210	107	0.794	0.883	0.836
<b>development sets, all filtering heuristics</b>						
Pouliquen et al. (2004)	637	781	278	0.449	0.696	0.546
none	782	115	133	0.872	0.855	0.863
geo-context	781	125	134	0.862	0.854	0.858
text	782	115	133	0.872	0.855	0.863
rel. importance	807	90	108	0.900	0.882	0.891
avg. distance	798	99	117	0.890	0.872	0.881
min. distance	798	99	117	0.890	0.872	0.881
all heuristics	808	98	107	0.892	0.883	0.887
<b>test sets, no filtering heuristics</b>						
Pouliquen et al. (2004)	540	1336	368	0.288	0.595	0.388
none	602	665	306	0.475	0.663	0.554
geo-context	602	804	306	0.428	0.663	0.520
text	602	665	306	0.475	0.663	0.554
rel. importance	682	585	226	0.538	0.751	0.627
avg. distance	619	648	289	0.489	0.682	0.569
min. distance	619	648	289	0.489	0.682	0.569
all heuristics	687	719	221	0.489	0.757	0.594
<b>test sets, all filtering heuristics</b>						
Pouliquen et al. (2004)	540	1336	368	0.288	0.595	0.388
none	583	256	325	0.695	0.642	0.667
geo-context	583	311	325	0.652	0.642	0.647
text	583	256	325	0.695	0.642	0.667
rel. importance	663	176	245	0.790	0.730	0.759
avg. distance	600	239	308	0.715	0.661	0.687
min. distance	600	239	308	0.715	0.661	0.687
all heuristics	668	226	240	0.747	0.736	0.741

Table A.4: Performance of the disambiguation heuristics for all languages with shallow-deep threshold 2.

	C	F	M	Precision	Recall	F-score
<b>development sets, heuristics from Pouliquen et al. (2004)</b>						
English	274	206	91	0.571	0.751	0.649
German	131	227	89	0.366	0.595	0.453
French	81	66	45	0.551	0.643	0.593
Spanish	115	234	31	0.330	0.788	0.465
Italian	36	48	22	0.429	0.621	0.507
<b>development sets, heuristics from this work</b>						
English	317	54	48	0.854	0.868	0.861
German	210	23	10	0.901	0.955	0.927
French	113	5	13	0.958	0.897	0.926
Spanish	119	6	27	0.952	0.815	0.878
Italian	49	10	9	0.831	0.845	0.838
<b>test sets, heuristics from Pouliquen et al. (2004)</b>						
English	227	557	101	0.290	0.692	0.408
German	103	277	108	0.271	0.488	0.349
French	61	48	80	0.560	0.433	0.488
Spanish	71	210	42	0.253	0.628	0.360
Italian	78	244	37	0.242	0.678	0.357
<b>test sets, heuristics from this work</b>						
English	257	67	71	0.793	0.784	0.788
German	152	58	59	0.724	0.720	0.722
French	92	26	49	0.780	0.652	0.710
Spanish	78	36	35	0.684	0.690	0.687
Italian	89	39	26	0.695	0.774	0.733

Table A.5: The performance of the heuristics for different languages.

All filtering and disambiguation heuristics are enabled, shallow-deep threshold is 2.

	<b>C</b>	<b>F</b>	<b>M</b>	<b>Precision</b>	<b>Recall</b>	<b>F-score</b>
<b>heuristics from Pouliquen et al. (2004)</b>						
Turkey	331	380	117	0.466	0.739	0.571
Chechnya	306	401	161	0.433	0.655	0.521
Reagan	297	1159	158	0.204	0.653	0.311
Iran	243	177	210	0.579	0.536	0.557
development sets	637	781	278	0.449	0.696	0.546
test sets	540	1336	368	0.288	0.595	0.388
all sets	1177	2117	646	0.357	0.646	0.460
<b>heuristics from this work</b>						
Turkey	431	23	17	0.949	0.962	0.956
Chechnya	377	75	90	0.834	0.807	0.820
Reagan	327	217	128	0.601	0.719	0.655
Iran	341	9	112	0.974	0.753	0.849
development sets	808	98	107	0.892	0.883	0.887
test sets	668	226	240	0.747	0.736	0.741
all sets	1476	324	347	0.820	0.810	0.815

Table A.6: The performance of the heuristics for different topics.

All filtering and disambiguation heuristics are enabled, shallow-deep threshold is 2.

# Bibliography

- Bilhaut, F., T. Charnois, P. Enjalbert, and Y. Mathet (2003). Geographic reference analysis for geographic document querying. In A. Kornai and B. Sundheim (Eds.), *Workshop on the Analysis of Geographic References at the NAACL-HLT 2003 conference*, Edmonton, Canada.
- Bronstein, I. N., K. A. Semendjajew, G. Musiol, and H. Mühlig (1999). *Taschenbuch der Mathematik* (4 ed.). Frankfurt am Main, Thun: Verlag Harri Deutsch.
- Brown Jr., W. E. and C. E. Botero (1997). The Florida Newspaper Project - Cuban Exile Newspapers at the University of Miami. From University of Florida, George A. Smathers Libraries. Available from: <http://www.uflib.ufl.edu/flnews/cuban.html> [Accessed 2004-08-23].
- Chamberlain, B. (1997). GIS FAQ, question 5.1: What is the best way to calculate the distance between 2 points? Available from: <http://www.census.gov/cgi-bin/geo/gisfaq?Q5.1> [Accessed 2004-08-23].
- Chinchor, N. (1997). MUC-7 Named Entity Task Definition. In *Message Understanding Conference Proceedings: MUC 7*.
- Densham, I. and J. Reid (2003). A geo-coding service encompassing a geo-parsing tool and integrated digital gazetteer service. In A. Kornai and B. Sundheim (Eds.), *Workshop on the Analysis of Geographic References at the NAACL-HLT 2003 conference*, Edmonton, Canada.
- Ehrlich, D., T. De Groot, C. Louvrier, and B. Eckhardt (2003). Digital Map Archive - An information infrastructure based on location. Technical Report JRC Technical Note I.03.10, Joint Research Centre.
- Eisenberg, J. D. (2002). *SVG Essentials*. O'Reilly and Associates.
- Fatman, S. and A. Stewart-Noble (2000). Llanfairpwllgwyngyllgogerychwyrndrobwlllantysiliogogoch, Anglesey, Wales. From BBC, h2g2 - The Guide to Life, The

- Universe and Everything. Available from: <http://www.bbc.co.uk/h2g2/guide/A403642> [Accessed 2004-08-23].
- Gale, W. A., K. W. Church, and D. Yarowsky (1992). One Sense per Discourse. In *Proceedings of the ARPA Workshop on Speech and Natural Language Processing*, 233–237.
- Ignat, C., B. Pouliquen, A. Ribeiro, and R. Steinberger (2003). Extending an Information Extraction Tool Set to Central and Eastern European Languages. In H. Cunningham, E. Paskaleva, K. Bontcheva, and G. Angelova (Eds.), *Proceedings of the International Workshop Information Extraction for Slavonic and Other Central and Eastern European Languages*, 33–39. Borovets, Bulgaria.
- Leidner, J. L., G. Sinclair, and B. Webber (2003). Grounding spatial named entities for information extraction and question answering. In A. Kornai and B. Sundheim (Eds.), *Workshop on the Analysis of Geographic References at the NAACL-HLT 2003 conference*, Edmonton, Canada, 31–38.
- Li, H., R. K. Srihari, C. Niu, and W. Li (2003). InfoXtract location normalization: a hybrid approach to geographic references in information extraction. In A. Kornai and B. Sundheim (Eds.), *Workshop on the Analysis of Geographic References at the NAACL-HLT 2003 conference*, Edmonton, Canada, 39–44.
- Makhoul, J., F. Kubala, R. Schwartz, and R. Weischedel (1999). Performance Measures For Information Extraction. In *Proceedings of the DARPA Broadcast News Workshop*, Herson, VA, 249–252.
- Mikheev, A., M. Moens, and C. Grover (1999). Named Entity Recognition without Gazetteers. In *Proceedings of EACL '99*, 1–8. Bergen, Norway.
- Oakes, M. P. (1998). *Statistics for Corpus Linguistics*. Edinburgh Textbooks in Empirical Linguistics. Edinburgh: Edinburgh University Press.
- Pearson, F. (1990). *Map Projections: Theory and Applications*. Boca Radon, Florida: CRC Press.
- Pouliquen, B., R. Steinberger, and C. Ignat (2004). Automatic Linking of Similar Texts Across Languages. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov (Eds.), *Current Issues in Linguistic Theory - Recent Advances in Natural Language Processing III*, Amsterdam. John Benjamins Publishers.
- Pouliquen, B., R. Steinberger, C. Ignat, and T. De Groeve (2004). Geographical Information Recognition and Visualisation in Texts Written in Various Languages. In

- Proceedings of the 19th Annual ACM Symposium on Applied Computing*, Nicosia, Cyprus, 1051–1058.
- Rivero Rojas, J. G. (1999). Bibliometric Analyses of Spanish Language Journals: Bradford, Lotka and Zipf. In C. Macias-Chapula (Ed.), *Proceedings of the Seventh Conference of the International Society for Scientometrics and Informetrics*, Colima, Mexico. Available from: <http://www.udlap.mx/~riverog/grr/issi7.htm> [Accessed 2004-08-23].
- Scheurich, J. (2001). LinuxTag 2001 proceedings: VRML/X3D (Virtual Reality Modeling Language) unter Unix/Linux: 3D-Welten im WWW. From Universität Stuttgart, Institut für Computeranwendungen / Abteilung Computersimulation und Visualisierung. Available from: <http://www.csv.ica.uni-stuttgart.de/vrml/linuxtag/> [Accessed 2004-08-23].
- Sinnott, R. W. (1984). Virtues of the Haversine. *Sky and Telescope* 68(2), 159.
- Tjong Kim Sang, E. F. and F. De Meulder (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In W. Daelemans and M. Osborne (Eds.), *Proceedings of CoNLL-2003*, 142–147. Edmonton, Canada.
- Tomlin, C. D. (1990). *Geographic Information Systems and Cartographic Modeling*. Englewood Cliffs, New Jersey: Prentice-Hall.
- van Rijsbergen, C. J. (1979). *Information Retrieval*, Volume 2. Dept. of Computer Science, University of Glasgow.
- W3C SVG Working Group (Ed.) (2003). Scalable Vector Graphics (SVG) 1.1 Specification. Available from: <http://www.w3.org/TR/SVG/index.html> [Accessed 2004-08-23].
- Web3D Consortium (Ed.) (2002). The Virtual Reality Modeling Language ISO/IEC 14772. Available from: <http://www.web3d.org/x3d/specifications/vrml/> [Accessed 2004-08-23].
- Web3D Consortium (Ed.) (2003). Extensible 3D (X3D) Specification, ISO/IEC FDIS 19775:200x. Available from: <http://www.web3d.org/x3d/specifications/ISO-IEC-19775-FDIS-X3dAbstractSpecification/> [Accessed 2004-08-23].
- Weisstein, E. W. (1999a). Inverse Cotangent. From MathWorld—A Wolfram Web Resource. Available from: <http://mathworld.wolfram.com/InverseCotangent.html> [Accessed 2004-08-23].

- Weisstein, E. W. (1999b). Inverse Function. From MathWorld—A Wolfram Web Resource. Available from: <http://mathworld.wolfram.com/InverseFunction.html> [Accessed 2004-08-23].
- Wikipedia (2004a). Bourbon County, Kentucky. From Wikipedia: The Free Encyclopedia. Available from: [http://en.wikipedia.org/wiki/Bourbon\\_County%2C\\_Kentucky](http://en.wikipedia.org/wiki/Bourbon_County%2C_Kentucky) [Accessed 2004-08-23].
- Wikipedia (2004b). Grozny. From Wikipedia: The Free Encyclopedia. Available from: <http://en.wikipedia.org/wiki/Grozny> [Accessed 2004-08-23].
- Wikipedia (2004c). Open source intelligence. From Wikipedia: The Free Encyclopedia. Available from: [http://en.wikipedia.org/wiki/Open\\_source\\_intelligence](http://en.wikipedia.org/wiki/Open_source_intelligence) [Accessed 2004-08-23].
- Zipf, G. K. (1949). *Human behavior and the principle of least effort : an introduction to human ecology*. Cambridge, Mass.: Cambridge University Press.

## Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde nach meiner besten Kenntnis bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ispra, den 23. August 2004

Unterschrift

---