Towards a search system for the Web exploiting spatial data of a web document

Stefan Dlugolinsky, Michal Laclavik, Ladislav Hluchy

Institute of Informatics, Slovak Academy of Sciences

Introduction

- Full text search services like
  - Google
  - Yahoo!
  - MSN
  - ...

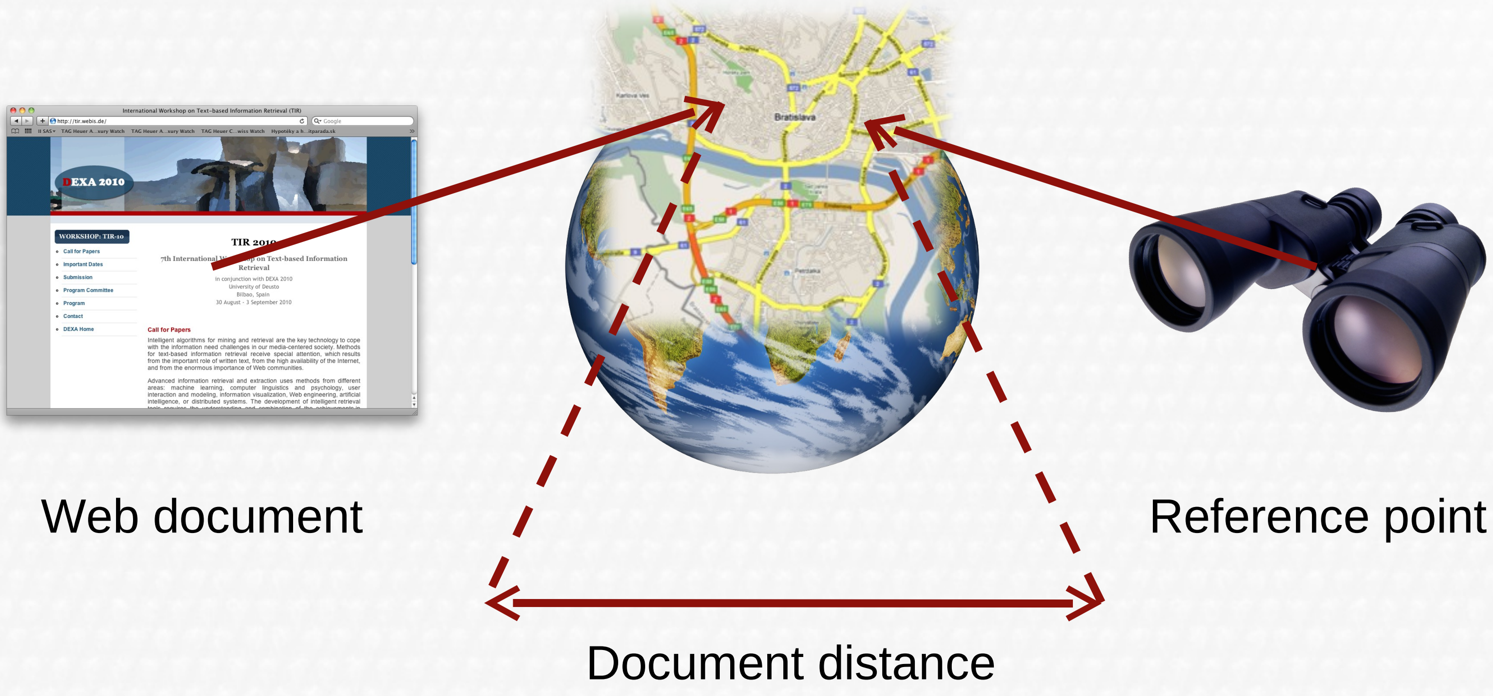
  determine the relevance of the web document by its content and importance (PageRank).

- Sometimes, we would like to prefer results, whose relevance is determined by a distance from us, where the distance is measured from the business location presented in the web document.

- Examples:
  - We are going to visit Bilbao and we are looking for a hotel.
  - We are looking for the nearest shopping centre, restaurant, pharmacy, etc.

Introduction

- **Geographic distance of a web document**
  Under a document's geographic distance, we understand the distance between a search reference point and a geographic location related to the document.



TIR 2010

7th International W[illegible]p on Text-based Information Retrieval

in conjunction with DEXA 2010
University of Deusto
Bilbao, Spain
30 August - 3 September 2010

Call for Papers

Intelligent algorithms for mining and retrieval are the key technology to cope with the information need challenges in our media-centered society. Methods for text-based information retrieval receive special attention, which results from the important role of written text, from the high availability of the Internet, and from the enormous importance of Web communities.

Advanced information retrieval and extraction uses methods from different areas: machine learning, computer linguistics and psychology, user interaction and modeling, information visualization, Web engineering, artificial intelligence, or distributed systems. The development of intelligent retrieval

Introduction

- Present search services like

1. Google Maps http://maps.google.com/
2. Yahoo Maps http://maps.yahoo.com/
3. ZlateStranky.sk (Slovak Yellow pages) http://www.zlatestranky.sk/
4. Kompas.sk http://mapy.zoznam.sk/
5. MAPY.SK http://mapy.atlas.sk/
6. Umkreisfinder http://www.umkreisfinder.de/

search documents by distance, **BUT**

– the search domain is created manually by users (1, 2), or built from the content of service catalogs – registered companies (3, 4)

– some search services (5) search only within a set of geographic object names (street names, city names, ...)

– geographic location must be explicitly set by special HTML meta tag, or is known by an URLof the document (6)

Introduction

- Our goal was to create a search service which could:
  - automatically create its search domain (web documents)
  - automatically determine geographic positions related to web documents
  - search web documents by geographic distance according to search query
- Above tasks rose up main problems, which should be treated:
  - Assigning geographic locations to a web document
    - Extraction of geographic entities from the documents
    - Geocoding extracted entities
  - Indexing documents by related geographic positions
  - Search by distance

The core of the system

- We used NUTCH search engine as the basis of our search system, which:
  - is open source and platform independent (Java)
  - is built on LUCENE text search engine library – an implementation of high-performance indexing and searching
  - includes web crawler for crawling web documents
  - includes parsers for different types of documents (HTML, PDF, RTF, ...)
  - except LUCENE, it has also integrated HADOOP, an open-source implementation of frameworks for reliable, scalable, distributed computing and data storage
  - has a plug-in system for easy extension

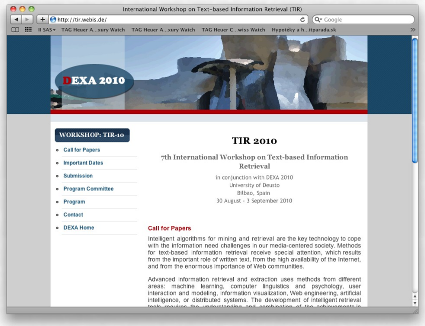
DistanceSearch plug-in

- We have built the search system extending NUTCH by our plug-in **DistanceSearch** and also by customizing the user interface.

- The **DistanceSearch** plug-in deals with three main issues:
  - **Extraction** of geographic coordinates from web documents and geocoding
  - **Indexing** by geographic distance
  - **Search** by geographic distance

Determining the geographic location of the document

- Strategy:
  - Extract all the possible geographic entities from **the textual content of the document**:
    - GPS coordinates (more accurate, no geocoding)
    - Postal addresses (less accurate, requires geocoding)
  - Geocode all the found street addresses to geographic coordinates
- Extracted coordinates are the input for the indexing process



TIR 2010

7th International Workshop on Text-based Information Retrieval

in conjunction with DEXA 2010
University of Deusto
Bilbao, Spain
30 August - 3 September 2010

Call for Papers

Intelligent algorithms for mining and retrieval are the key technology to cope with the information need challenges in our media-centered society. Methods for text-based information retrieval receive special attention, which results from the important role of written text, from the high availability of the Internet, and from the enormous importance of Web communities.

Advanced information retrieval and extraction uses methods from different areas: machine learning, computer linguistics and psychology, user interaction and modeling, information visualization, Web engineering, artificial intelligence, or distributed systems. The development of intelligent retrieval

GPS coords.
Addresses

Latitude and Longitude
48.1668595, 17.1032454
48.1441842, 17.1095594
48.1099847, 17.1146532

Geographic entity extraction

- Extraction model
  - Based on **regular expressions (regexes)**
  - Extracted entities are in the form of **key-value** pairs
    - for example. "sk.address.street.name" => "Jasovska”
  - We define extraction patterns by custom meta-language in XML
    - simple modification of extraction patterns
    - easily expandable with new extraction patterns
    - extraction model supports macros and naming of regex back references (it is easier to create complex patterns and reuse them in other patterns)

An example of address extraction pattern

```
<pattern name=“Postcode" class=“Postcode">
  <regexp><![CDATA[[0-9]{3}\s*[0-9]{2}]]></regexp>
</pattern>

<pattern name=“Word">
  <regexp><![CDATA[(\p{Lu}\p{Ll}*|\p{Ll}+)]]></regexp>
</pattern>

<pattern name=“Name">
  <regexp><![CDATA[(\p{Lu}\p{Ll}+)]]></regexp>
</pattern>

<pattern name=“StreetName" class=“StreetName">
  <regexp><![CDATA[\p{#Name}\.?( +\p{#Word}\.?)?( +\p{#Word}\.?)?( +\p{#Word}\.?)?]]></regexp>
</pattern>

<pattern name=“BuildingNumber" class=“BuildingNumber">
  <regexp><![CDATA[([1-9][0-9]{0,3} *[/-] *)?([1-9]|[1-9][0-9])(/?[a-zA-Z])?]]></regexp>
</pattern>

<pattern name=“CityName" class=“CityName">
  <regexp><![CDATA[\p{#Name}( +\p{#Name})?]]></regexp>
</pattern>

<pattern class=“Address">
  <regexp><![CDATA[\p{#CityName} +\p{#BuildingNumber}(,\s*|\s+)\p{#Postcode}\s+\p{#CityName}]]></regexp>
</pattern>
```

Macro reference

regex

macros

pattern

Geographic entity extraction

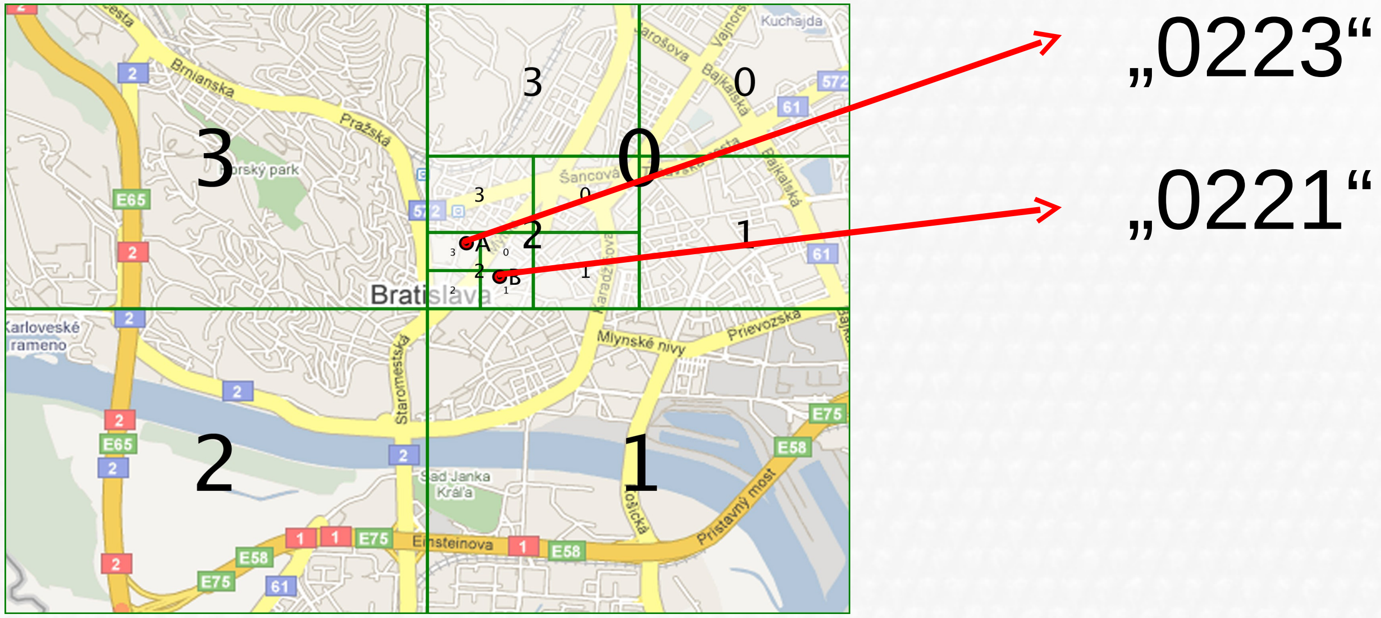
- Extraction patterns:
  - 2 extraction patterns for Slovak postal address extraction. Patterns deal with the diversity of city and street naming, also building numbering:
    - (<StreetName> <BuildingNumber>?)?<Postcode> <CityName>
    - (<StreetName> <BuildingNumber>?)?<CityName> <Postcode>
  - 4 extraction patterns for GPS coordinate extraction

Geocoding of postal addresses

- **Geocoding**
  a process of converting postal addresses or known geographic object names (landmarks, institutions, etc.) into geographic coordinates.
- The original idea was to create our own geocoding service using **gazetteer**.
  - We designed, implemented and verified data model of the gazetteer, filled it with addresses and got hierarchical structure of related street names, postcodes, city names, county names and region names, according to Slovak address format
  - However, we did not have very precise geographic data. Therefore we were able to geocode addresses only up to the level of street names (without exploiting extracted building numbers, what would be more precise)
- For the reason to be more precise in geocoding, we have decided to use **external geocoding services**
  - Google
  - Yahoo
- We cache all calls to geocoding services into the database (it is due to each service request limits and recrawling optimalization)

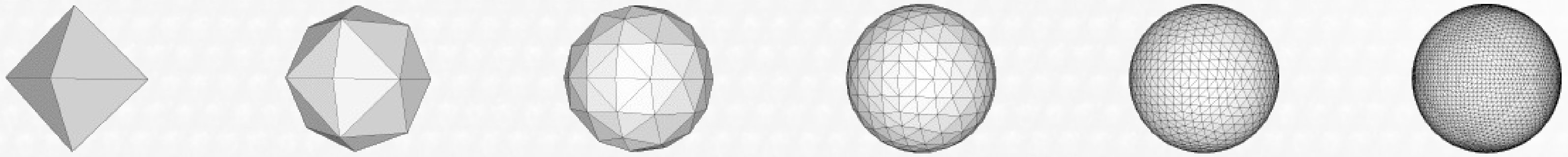
Indexing

- We wanted to index web document by **more than just a one geographic location** - the problem: LUCENE's indexing capabilities inside NUTCH (supports only string values) and tightness between indexing and searching (need to be fast)
- Efforts to represent latitude and longitude in one string, a hash.
- The idea:



Indexing

- Hierarchical Triangular Mesh method (http://www.skyserver.org/htm/index.html)
  - Developed with a grant support of NASA (for use in digital astronomical maps).
  - Built on the same principle, but divides the spherical surface into the triangles:



  - The precision of HTM index grows with the level of triangle subdivision (the resolution at the level 25 is about 0.6 m)
  - We use HTM index to index document by its related geographic coordinates.
    - An example: 48.1668595 17.1032454 => N3313312100112100303

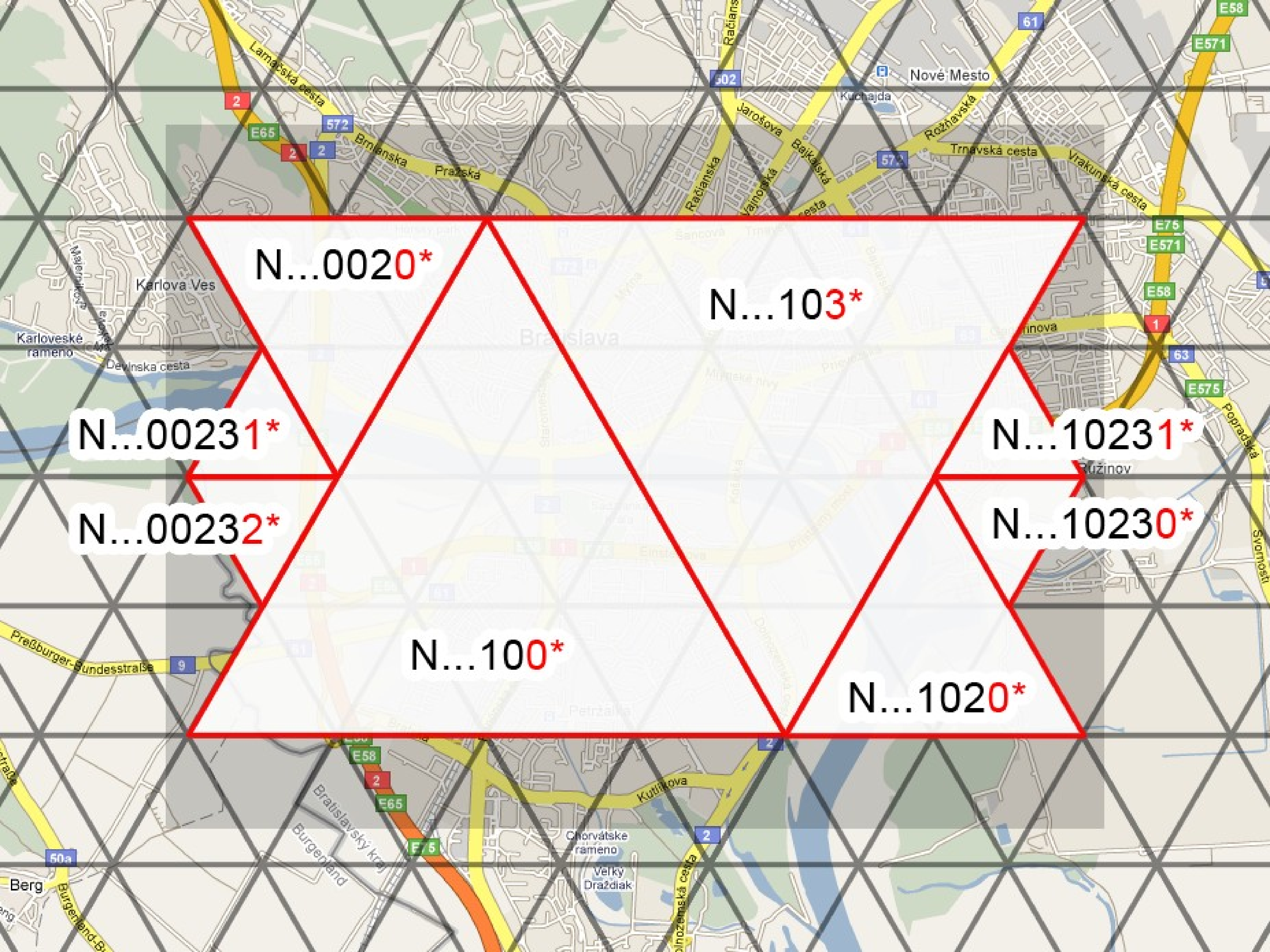
Feature: If there are two points with common prefixes, they are close to each other. The more longer prefix they share, the closer they lie to each other.

Search

- The principle of search:

  Extension of the **generated NUTCH query** by an **area search query** based on HTM region prefixes.

1. A set of common **HTM prefixes** is computed for triangles within the search area.


N…0020*
N…103*
N…00231*
N…00232*
N…10231*
N…10230*
N…100*
N…1020*
Bratislava
Karlova Ves
Nové Mesto
Ružinov
Petržalka
Berg

Search

2. There is a **prefix subquery** created for each computed common HTM prefix:

   PK_i = PrefixQuery("htm", $prefix_i$)

3. Prefix subqueries are joined into an **area search query**:

   (PK_1 ***OR*** PK_2 ***OR*** ... PK_n)

4. The area search query is then added to the generated NUTCH query:

   (NUTCH query) ***AND*** (PK_1 *OR* PK_2 *OR* ... PK_n)

xtended Nutch query is fulfilled for each document, which is relevant to original query and “lies” in the search region.

Evaluation of address extraction

- Set of test web documents
  - Google catalogue, section “Bratislava” and Slovakia”

Documents	3183
Relevant documents	380
Addresses in documents	847
Correct address extractions	741
False address extractions	730
Missed addresses	106
Precision	50.37%
Recall	87.49%

$$Precision = \frac{|R \cap I|}{|I|} = \frac{741}{1471} = 0.5037 = 50.37\%$$

$$Recall = \frac{|R \cap I|}{|R|} = \frac{741}{847} = 0.8749 = 87.49\%$$

Distributed crawling and its results

- We installed crawler on a cluster with 8 nodes, each with the following configuration:

Processor	Intel® Core™ 2 Quad CPU Q9550 @ 2.83GHz
RAM	4 GB
HDD	WDC WD7500AACS-0 (750 GB)
OS	Linux 2.6.24-19-generic X86_64 GNU/Linux

- We have configured Hadoop to simultaneously run 8 jobs on every node, i.e. 64 parallel jobs on whole cluster.

Crawling start:

1. http://www.zoznam.sk/
2. http://www.centrum.sk/
3. http://www.best.sk/
4. http://szm.sk/
5. http://www.atlas.sk/
6. http://www.katalog.sk/
7. http://www.azet.sk/

max. level 10
limit 100 000 / level

Crawling time	20h 46m 7s
Crawled data size (index + cache)	8.3 GB
Index size	535.7 MB
Documents crawled	408 096
Documents without entity extraction	349 561 (85.66 %)
Documents with entity extraction	58 535 (14.34 %)
Extracted entities	245 673
Geo-coding requests	21 463
Average number of entities per document	0.60
Recognized GPS coordinates	256
Not geo-coded address extractions	123 731
Geo-coded address extractions	121 686

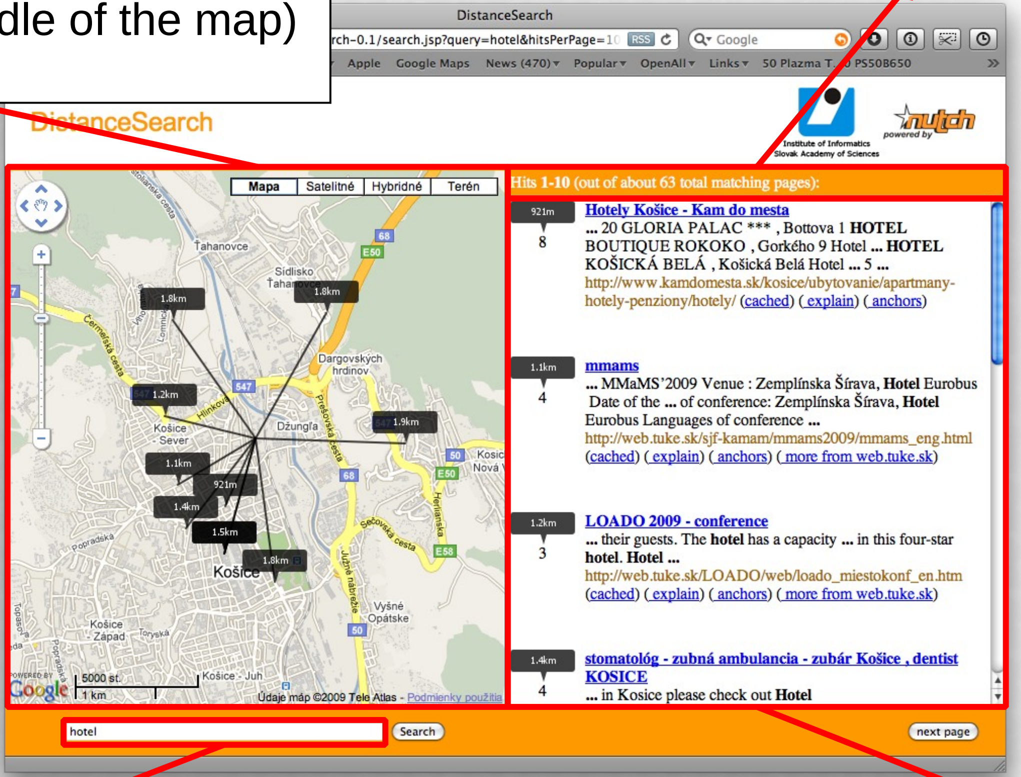
Comparison with similar services

	DistanceSearch (NUTCH)	GeoPosition (NUTCH)	Google Maps	Yahoo Local Maps	Kompas.sk	ZlateStranky.sk
Search domain	web	web	db	db	db	db
Geo-entity extraction from document textual content	✓	✗	✗	✗	✗	✗
Search in map viewport	✓	✗	✓	✓	✗	✓
Display results on a map	✓	✗	✓	✓	✓	✓
Order results by distance	✓	✗	✗	✗	✗	✓

Searcher user interface

Map viewport
- defines search area (reference point is in the middle of the map)
- display of results

Search results details


DistanceSearch
rch-0.1/search.jsp?query=hotel&hitsPerPage=10
Apple Google Maps News (470) Popular OpenAll Links 50 Plazma T. PS50B650
DistanceSearch
Institute of Informatics Slovak Academy of Sciences
powered by nutch
Mapa Satelitné Hybridné Terén
Košice
921m
1.1km
1.2km
1.4km
1.5km
1.8km
1.9km
hotel
Search
next page

Hits **1-10** (out of about 63 total matching pages):

921m 8 — [Hotely Košice - Kam do mesta]
... 20 GLORIA PALAC *** , Bottova 1 **HOTEL** BOUTIQUE ROKOKO , Gorkého 9 Hotel ... **HOTEL** KOŠICKÁ BELÁ , Košická Belá Hotel ... 5 ...
http://www.kamdomesta.sk/kosice/ubytovanie/apartmany-hotely-penziony/hotely/ (cached) (explain) (anchors)

1.1km 4 — [mmams]
... MMaMS'2009 Venue : Zemplínska Šírava, **Hotel** Eurobus Date of the ... of conference: Zemplínska Šírava, **Hotel** Eurobus Languages of conference ...
http://web.tuke.sk/sjf-kamam/mmams2009/mmams_eng.html (cached) (explain) (anchors) (more from web.tuke.sk)

1.2km 3 — [LOADO 2009 - conference]
... their guests. The **hotel** has a capacity ... in this four-star **hotel**. **Hotel** ...
http://web.tuke.sk/LOADO/web/loado_miestokonf_en.htm (cached) (explain) (anchors) (more from web.tuke.sk)

1.4km 4 — [stomatológ - zubná ambulancia - zubár Košice , dentist KOSICE]
... in Kosice please check out **Hotel**

Search query input field

List of results
- results are ordered by distance from the reference point

Conclusion

- Advantages:
  - Automatic address extraction and geocoding with high recall
  - Address and GPS extraction patterns are easily extensible
  - Documents are indexed by more than just a one related geographic location
  - Distributed crawling+extraction+indexing
  - External geocoding services can be used
  - Caching geocoding calls (decreases direct calls to geocoding services when recrawling)
  - Results are shown on a map
- Disadvantages and future work:
  - Lower extraction precision, but new patterns can improve it. Currently we have integrated also GATE (General Architecture for Text Extraction) in Nutch and trying it.
  - Geographic entity extraction does not handle with microformats, RDF in XHTML, embeded maps (Google Maps, Yahoo Local Maps, ...)
  - Geographic positions are associated only with their parent pages, but sometimes they could be associated with pages linked to their parent page (a future work challenge)

Thank you

DistanceSearch plug-in

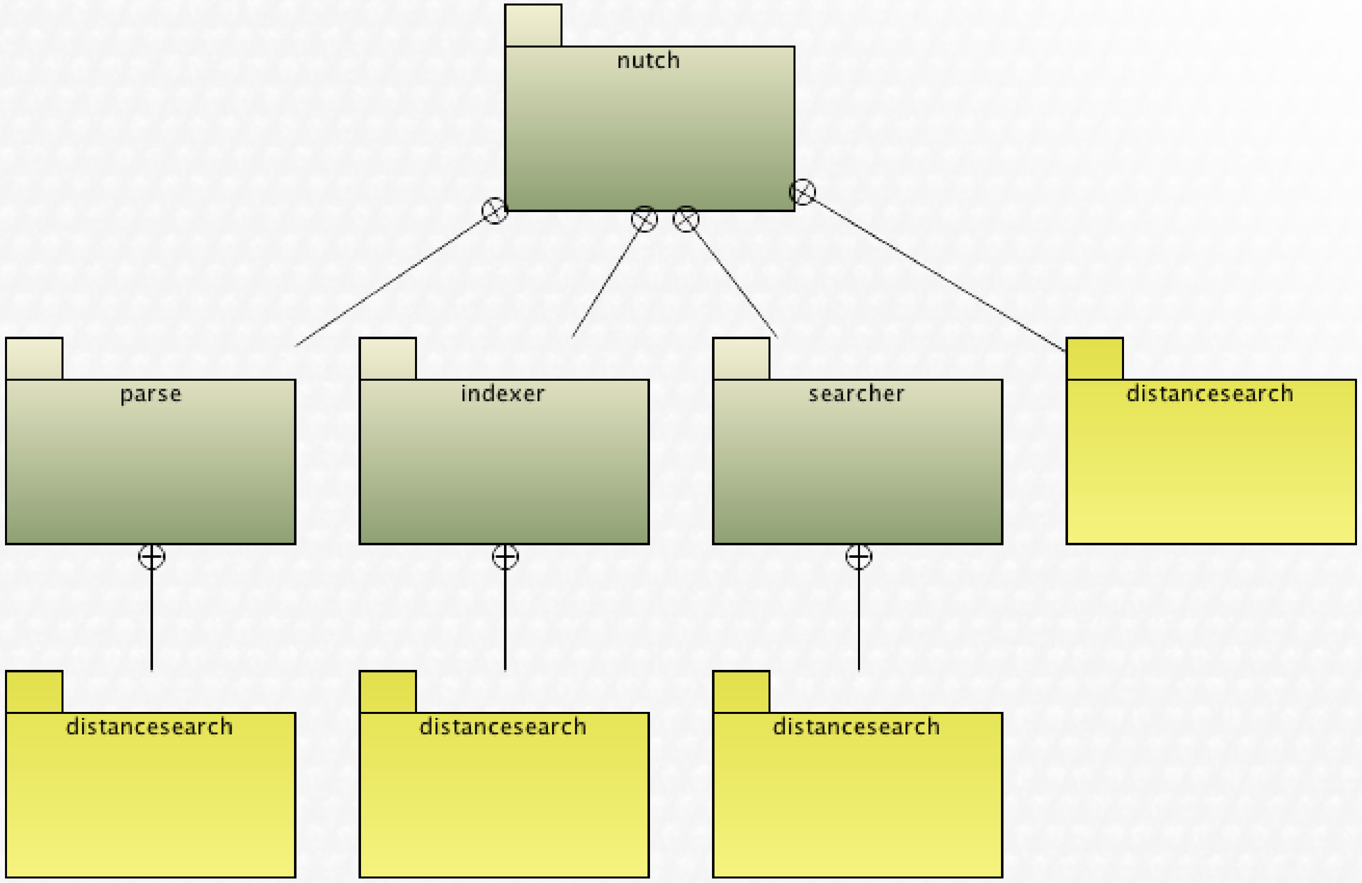
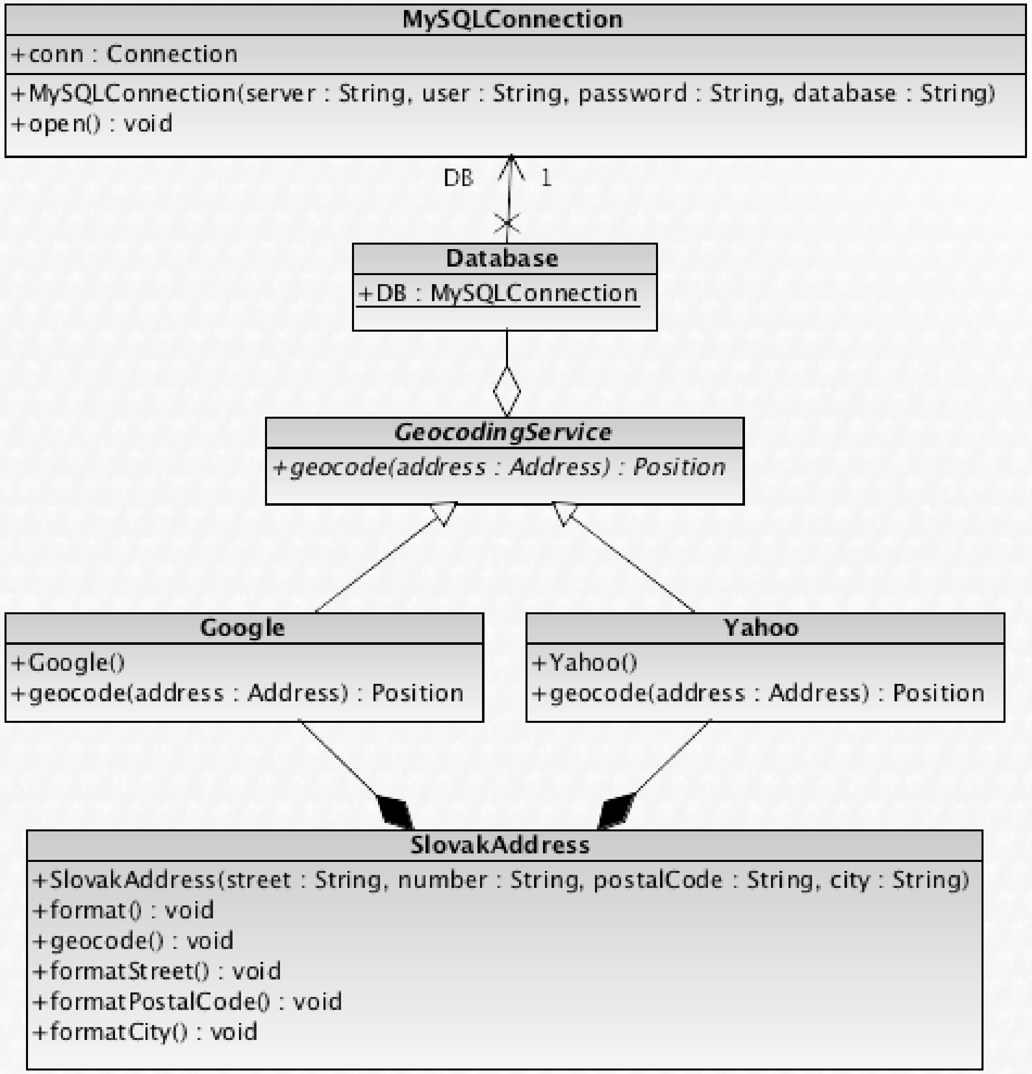


Diagram of DistanceSearch plug-in integration in NUTCH

Geocoding of postal addresses


MySQLConnection
+conn : Connection
+MySQLConnection(server : String, user : String, password : String, database : String)
+open() : void
DB
1
Database
+DB : MySQLConnection
GeocodingService
+geocode(address : Address) : Position
Google
+Google()
+geocode(address : Address) : Position
Yahoo
+Yahoo()
+geocode(address : Address) : Position
SlovakAddress
+SlovakAddress(street : String, number : String, postalCode : String, city : String)
+format() : void
+geocode() : void
+formatStreet() : void
+formatPostalCode() : void
+formatCity() : void