LoCloud

Local content in a Europeana cloud

D3.3: Metadata Enrichment services

Authors:
Aitor Soroa(EHU)
Arantxa Otegi(EHU)
Eneko Agirre(EHU)
Rodrigo Agerri (EHU)

Version: 1.0

LoCloud is funded by the European Commission's
ICT Policy Support Programme

cip competitiveness and innovation framework programme 2007–2013

Version	Date	Author	Organisation	Description
0.1	April 2014	Rodrigo Agerri	EHU	First version of NED systems
0.2	June 2014	Arantxa Otegi	EHU	Add section on evaluation
0.3	August 2014	Aitor Soroa	EHU	First complete draft
0.4	September 2014	Aitor Soroa	EHU	Final version - Published

View the LoCloud project deliverables

Statement of originality:
This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Contents

Executive summary

Metadata enrichment services automatically link Cultural Heritage (CH) items with external information such as DBpedia pages or vocabulary concepts. Metadata enrichment is a useful task which assists users by providing contextual information about a particular item. It also helps in the task of categorizing CH items with relevant terms and concepts from a particular vocabulary. Metadata enrichment comprises two different micro-services:

- **Background link** micro-service, which automatically links CH items to background information contained in pages from an external resource like Wikipedia or DBpedia.
- **Vocabulary matching** micro-service, which automatically links metadata relevant to relevant classes in a provided vocabulary.

The background link micro-service relies on DBpedia Spotlight, a state-of-the-art tool for performing Named Entity Disambiguation (NED). DBpedia Spotlight was chosen because it performed best in our experiments, as shown in the deliverable.

The vocabulary matching micro-service is developed from scratch for the project. It synchronizes with the *vocabulary service*[1], a collaborative platform to explore the potentials of crowdsourcing as a way of developing multilingual, semantic thesauri for local heritage content.

All micro-services are implemented as REST services and are deployed into virtual machines (VM). Documentation on the service, including full API specification, is avaliable at this address:

http://support.locloud.eu/LoCloud%20Enrichment%20Microservice

The metadata enrichment micro-services are used in the various enrichment workflows automatically through the LoCloud Generic Enrichment Service, which allows to orchestrate various REST micro-services into complex enrichment workflows.
The purpose of this deliverable is to describe each metadata enrichment micro-service and to specify their corresponding APIs.

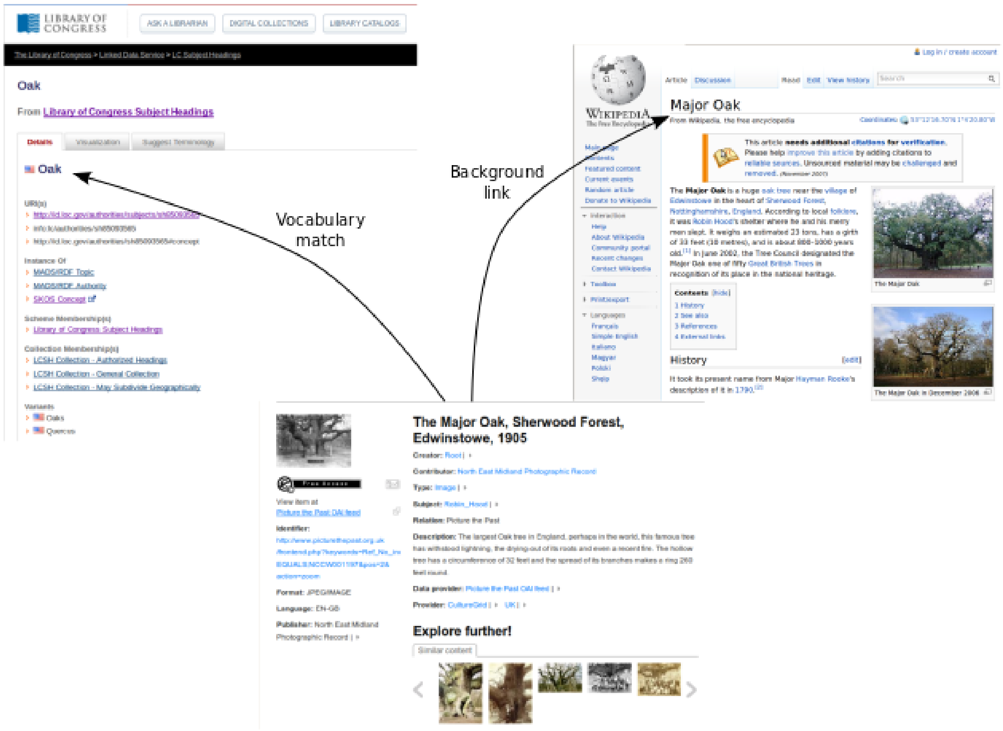
[1] Task 3.4 of LoCloud project.

1 Introduction

This document describes the services developed within the LoCloud project with the aim of enriching the metadata information associated with Cultural Heritage (CH) items. Metadata enrichment is a useful task which assists users by providing contextual information about a particular item. It also helps in the task of categorizing CH items with relevant terms and concepts from a particular vocabulary. Metadata enrichment comprises two different micro-services:

- **Background link** micro-service, which automatically links CH items to background information contained in pages from an external resource like Wikipedia or DBpedia.
- **Vocabulary matching** micro-service, which automatically links metadata relevant to relevant classes in a provided vocabulary.

Figure 1 shows an example of a vocabulary match and background link for a particular Europeana CH item. The bottom of the figure shows the CH item describing "The Major Oak" as displayed by the Europeana portal[2]. The item has been enriched and linked to two different elements: the vocabulary concept "Oak" as defined by the Library of Congress Subject Headings vocabulary (left part of the figure), and a Wikipedia page describing the same tree (the right part).



Oak

From Library of Congress Subject Headings

Oak

URI(s)
- http://id.loc.gov/authorities/subjects/sh85093585
- info:lc/authorities/sh85093585
- http://id.loc.gov/authorities/sh85093585#concept

Instance Of
- MADS/RDF Topic
- MADS/RDF Authority
- SKOS Concept

Scheme Membership(s)
- Library of Congress Subject Headings

Collection Membership(s)
- LCSH Collection - Authorized Headings
- LCSH Collection - General Collection
- LCSH Collection - May Subdivide Geographically

Variants
- Oaks
- Quercus

Major Oak

From Wikipedia, the free encyclopedia

This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. (November 2007)

The **Major Oak** is a huge oak tree near the village of Edwinstowe in the heart of Sherwood Forest, Nottinghamshire, England. According to local folklore, it was Robin Hood's shelter where he and his merry men slept. It weighs an estimated 23 tons, has a girth of 33 feet (10 metres), and is about 800–1000 years old.[1] In June 2002, the Tree Council designated the Major Oak one of fifty Great British Trees in recognition of its place in the national heritage.

Contents [hide]
1 History
2 See also
3 References
4 External links

The Major Oak

History [edit]

It took its present name from Major Hayman Rooke's description of it in 1790.[2]

The Major Oak in December 2008

The Major Oak, Sherwood Forest, Edwinstowe, 1905

Creator: Root |
Contributor: North East Midland Photographic Record
Type: Image |
Subject: Robin_Hood |
Relation: Picture the Past
Description: The largest Oak tree in England, perhaps in the world, this famous tree has withstood lightning, the drying-out of its roots and even a recent fire. The hollow tree has a circumference of 32 feet and the spread of its branches makes a ring 260 feet round.
Data provider: Picture the Past OAI feed |
Provider: CultureGrid | UK |

View item at
Picture the Past OAI feed
Identifier:
http://www.picturethepast.org.uk/frontend.php?keywords=Ref_No_increment;EQUALS;NCCW001197&pos=2&action=zoom
Format: JPEG/IMAGE
Language: EN-GB
Publisher: North East Midland Photographic Record |

Figure 1: Example of a vocabulary match and background link for the Europeana item describing "The Major Oak" (at the bottom of the figure). The left part shows a match to a concept from the LCSH vocabulary. The right part shows a background link to a Wikipedia page.

We have implemented the following three micro-services for metadata enrichment:

- A background link micro-service for English.
- A background link micro-service for Spanish.

[2] http://www.europeana.eu/portal/record/2022323/C052AA1727D9C258801CF676473953A0861A47C0.html

- A multi-lingual vocabulary matching micro-service.

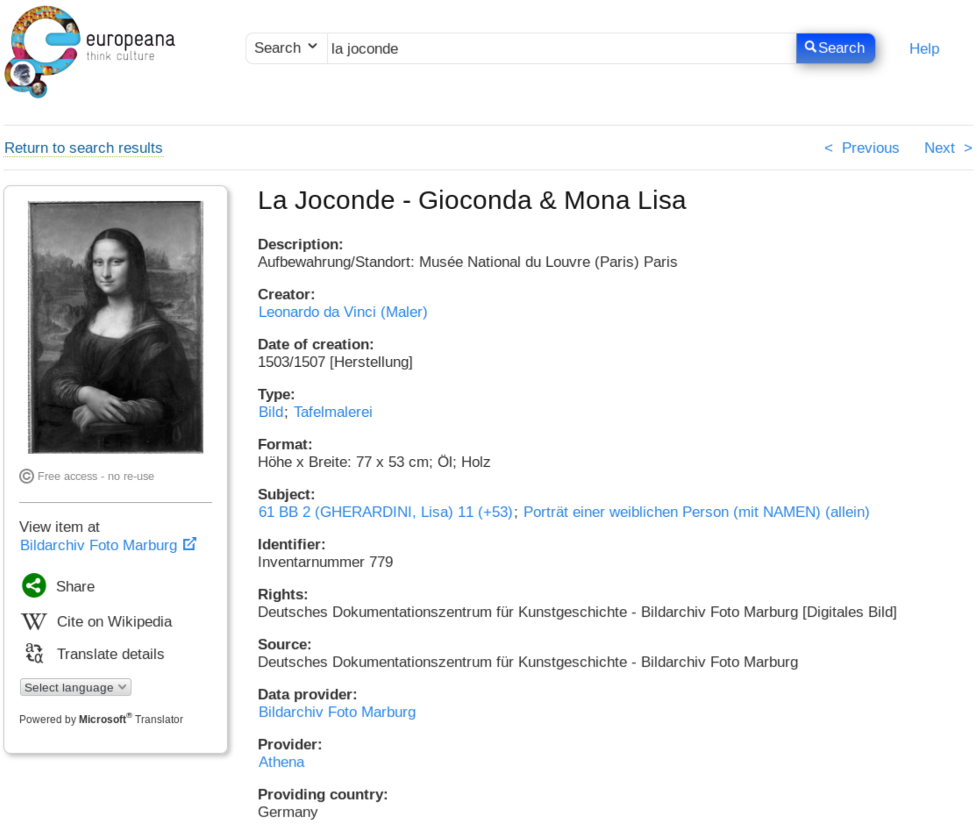
All micro-services are implemented as REST services and are deployed into virtual machines (VM). The micro-services are used in the various enrichment workflows automatically through the LoCloud Generic Enrichment Service. The generic enrichment service allows to orchestrate various REST micro-services into complex enrichment workflows. The user can create a workflow by selecting and combining the micro-services he wants. More specifically for metadata enrichment, the textual descriptions (e.g. a description element) is posted to the background link an vocabulary matching micro-services and is returned back enriched and integrated into an enriched version of EDM (Doerr et al., 2010).

The background link problem is closely related to the so called Named Entity Disambiguation (NED) task. Nowadays there exist many systems for performing NED, and therefore the micro-service has been implemented using such a NED tool. In the deliverable we perform a state-of-the art on NED tools and datasets, and present an evaluation of the chosen tool. The task of linking CH items with relevant vocabulary concepts has also gained much attention during recent years. However, to our knowledge there exist no tools nor standard datasets to test them. We thus designed and implemented the vocabulary matching micro-service from scratch.

The document is structured as follows. Section 2 describes in detail the background link micro-service. Section 2.2 and 2.3 provide an in-depth analysis of current state of the art systems for NED, including testing datasets. Section 2.4 reports the results when evaluating two of the best systems for English and Spanish. Section 3 describes the design of the vocabulary matching micro-service. Section 4 explains the actual implementation of both micro-services, and shows some examples of use. Section 6 describes the APIs for using the services. Finally, Section 7 draws some conclusions.

2 Background Link micro-service

Current efforts for the digitisation of Cultural Heritage are providing common users with access to vast amount of materials. Sometimes, though, this quantity comes at the cost of a restricted amount of metadata, with many items having very short descriptions and a lack of rich contextual information. For instance, figure 2 shows the "Mona Lisa" CH item as displayed in Europeana. The item does not provide rich information associated with the item. Wikipedia, in contrast, offers in-depth descriptions and links to related articles for many CH items, as shown in figure 3. Therefore, Wikipedia is a natural target for automatic enrichment of CH items.



La Joconde - Gioconda & Mona Lisa

Description:
Aufbewahrung/Standort: Musée National du Louvre (Paris) Paris

Creator:
Leonardo da Vinci (Maler)

Date of creation:
1503/1507 [Herstellung]

Type:
Bild; Tafelmalerei

Format:
Höhe x Breite: 77 x 53 cm; Öl; Holz

Subject:
61 BB 2 (GHERARDINI, Lisa) 11 (+53); Porträt einer weiblichen Person (mit NAMEN) (allein)

Identifier:
Inventarnummer 779

Rights:
Deutsches Dokumentationszentrum für Kunstgeschichte - Bildarchiv Foto Marburg [Digitales Bild]

Source:
Deutsches Dokumentationszentrum für Kunstgeschichte - Bildarchiv Foto Marburg

Data provider:
Bildarchiv Foto Marburg

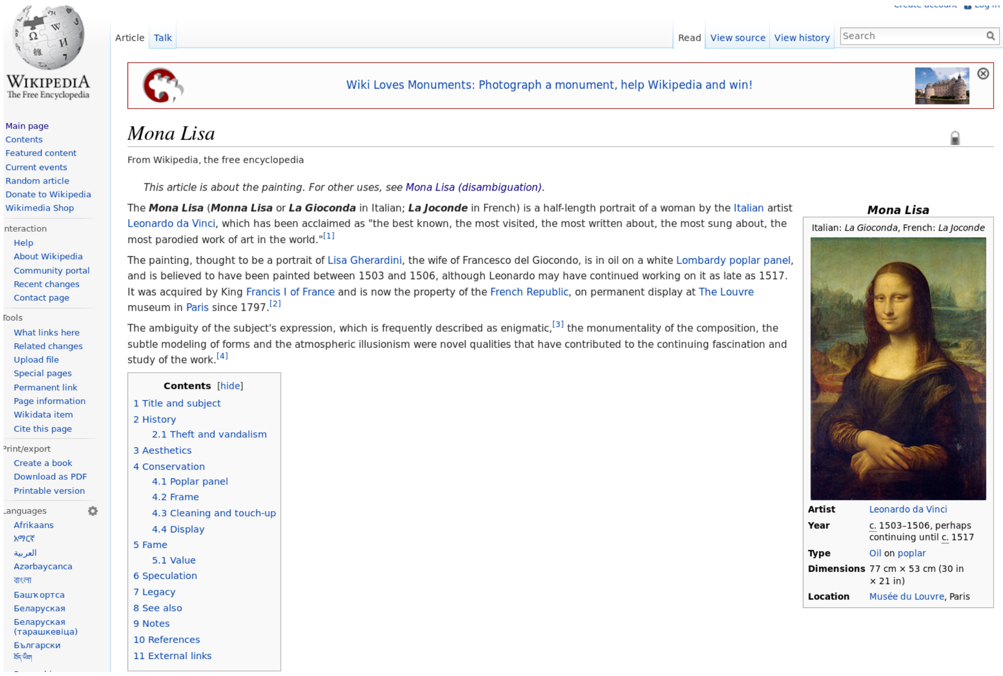
Provider:
Athena

Providing country:
Germany

Figure 2: Europeana CH item describing "Mona Lisa". Note that the description field does not provide any information about the artwork.

Enriching CH items with information from Wikipedia or other external resources is not novel. In Haslhofer et al. (2010), for instance, the authors also acknowledge the interest of enriching CH items. They present the LEMMO framework, a tool to help users annotate Europeana items with external resources (i.e. Web pages, Dbpedia entries, etc.), thus extending Europeana items with user-contributed annotations.

The goal of the background link micro-service implemented in LoCloud is to automatically enrich CH items with external elements. This task is closely related to the "Named Entity Disambiguation" (NED) problem. As will be shown below, nowadays there exist many wide-coverage NED tools which can be used to implement the background link micro-service. Unfortunately, to the best of our knowledge, none of the existing NED solution focuses on the CH domain. However, almost all existing NED tools link the text with encyclopedic resources such as Wikipedia or DBpedia, which cover many topics including CH. For instance, Wikipedia contains thousands of elements from the CH domain, including paintings, archeological items, artists, etc. Therefore, we decided to use such a general purpose NED tool for LoCloud.



Mona Lisa

From Wikipedia, the free encyclopedia

This article is about the painting. For other uses, see Mona Lisa (disambiguation).

The ***Mona Lisa*** (***Monna Lisa*** or ***La Gioconda*** in Italian; ***La Joconde*** in French) is a half-length portrait of a woman by the Italian artist Leonardo da Vinci, which has been acclaimed as "the best known, the most visited, the most written about, the most sung about, the most parodied work of art in the world."[1]

The painting, thought to be a portrait of Lisa Gherardini, the wife of Francesco del Giocondo, is in oil on a white Lombardy poplar panel, and is believed to have been painted between 1503 and 1506, although Leonardo may have continued working on it as late as 1517. It was acquired by King Francis I of France and is now the property of the French Republic, on permanent display at The Louvre museum in Paris since 1797.[2]

The ambiguity of the subject's expression, which is frequently described as enigmatic,[3] the monumentality of the composition, the subtle modeling of forms and the atmospheric illusionism were novel qualities that have contributed to the continuing fascination and study of the work.[4]

Contents [hide]

1 Title and subject
2 History
   2.1 Theft and vandalism
3 Aesthetics
4 Conservation
   4.1 Poplar panel
   4.2 Frame
   4.3 Cleaning and touch-up
   4.4 Display
5 Fame
   5.1 Value
6 Speculation
7 Legacy
8 See also
9 Notes
10 References
11 External links

Mona Lisa

Italian: *La Gioconda*, French: *La Joconde*

Artist	Leonardo da Vinci
Year	c. 1503–1506, perhaps continuing until c. 1517
Type	Oil on poplar
Dimensions	77 cm × 53 cm (30 in × 21 in)
Location	Musée du Louvre, Paris

Figure 3: Wikipedia entry for "Mona Lisa".

In this section we analyze various systems and data sources for NED. We first start with an in-depth survey of the current state-of-the-art, data sources, tools and technology related to NED. We then choose two open source NED systems, and evaluate them on standard data sources in both English and Spanish. As a result of this evaluation, the best tool for NED is chosen, so that it becomes the core element of the background link annotation micro-service.

2.1 Named Entity Disambiguation

The NED task focuses on recognizing, classifying and linking mentions of specific named entities in text. NED systems deal with two basic problems: name variation and name disambiguation. The former is produced because a named entity can be mentioned using a great variety of surface forms ("Mona Lisa", "La Gioconda", "Gioconda", etc.); the latter problem arises because the same surface form can refer to a variety of named entities: for example, the form "san juan" can be used to ambiguously refer to dozens of toponyms, persons, a saint, etc. (e.g, see http://en.wikipedia.org/wiki/San_Juan). Therefore, in order to provide an adequate and comprehensive account of named entities in text it is necessary to recognize the mention of a named entity, classify it as a type (e.g, person, location, painting, etc.), link it or disambiguate it to a specific entity, and resolve every form of mentioning to the same entity in a text.

The first task of any NED system is to detect and identify of specific entities in running text, a task called Named Entity Recognition and Classification (NERC). Current state-of-the-art processors achieve high performance in recognition and classification of general categories such as people, places, dates or organisations (Nadeau and Sekine, 2007), e.g. OpenCalais service for English[3].

Once the named entities are recognised they can be identified with respect to an existing catalogue. Wikipedia has become the *de facto* standard as such a named entity catalogue. *Entity Linking* is the process of automatic linking of the named entities occurring in free text to their corresponding Wikipedia articles. This task is typically

[3] http://www.opencalais.com/

regarded as a Word Sense Disambiguation (WSD) problem (Agirre and Edmonds, 2007), where Wikipedia provides both the dictionary and training examples. Public demos of systems which exploit Wikification (only for English) are Spotlight[4], CiceroLite from LCC[5] and, Zemanta[6], TAGME[7] or The Wiki Machine[8].

Automatic text Wikification implies solutions for NED (Mihalcea and Csomai, 2007). For unambiguous terms it is not a problem, but in other cases word sense disambiguation must be performed. For example, the Wikipedia disambiguation page lists many different articles that the term *oak tree* might refer to (a tree, a village in England, an artwork by Michael Craig-Martin, etc[9]).

The following sentence provides an example of a CH description with the corresponding Wikipedia links:

*The largest **Oak tree**[10] in **England**[11], perhaps in the world, this famous tree has withstood lightning.*

The named entity ambiguity problem has been formulated in two different ways. Within computational linguistics, the problem was first conceptualised as an extension of the coreference resolution problem (Bagga and Baldwin, 1998). The *Wikification* approach later used Wikipedia as a word sense disambiguation data set by attempting to reproduce the links between pages, as linked text is often ambiguous (Mihalcea and Csomai, 2007). Finally, using Wikipedia as in the Wikification approach, NERC was included as a preprocessing step and a link or NIL was required for all identified mentions (Bunescu and Pasca, 2006). This means that, as opposed to Wikification, links were to be provided only for named entities. The resulting terminology of these various approaches is cross-document coreference resolution, Wikification, and Named Entity Linking (NEL)[12]. The term Named Entity Disambiguation will be used to refer to any of these three tasks indistinctly (Hachey et al., 2012).

Different approaches have been proposed for the NEL. A system typically searches for candidate entities and then disambiguates them, returning either the best candidate or NIL. Thus, although both WSD and NEL address ambiguity and synonymy in natural language, there is an important difference. In NEL, it is not assumed that the KB is complete, as it is in WSD with respect to WordNet, which means that named entity mentions present in the text which do not have a reference in the KB must be marked as NIL. Moreover, the variation of named entity mentions is higher than that of lexical mentions in WSD, which makes the disambiguation process much noisier.

The rise to prominence of Wikipedia has allowed developing wide-coverage NEL systems. The most popular task, in which both datasets and NEL systems can be found, is the Knowledge Base Population (KBP) task at the NIST Text Analysis Conference (TAC). The goal of KBP is to promote research in automated systems that discover information about named entities as found in a large corpus and incorporate this information into a knowledge base. The TAC 2012 fields tasks in three areas all aimed at improving the ability to automatically populate knowledge bases from text. For our purposes the Entity-Linking task is the most relevant:

> "Given a name (of a Person, Organization, or Geopolitical Entity) and a document containing that name, determine the KB node for the named entity, adding a new node for the entity if it is not already in the KB. The reference KB is derived from English Wikipedia, while source documents come from a variety of languages, including English, Chinese, and Spanish".[13]

The popularity of the KBP task has led to a huge number of NEL systems, although given that every participant was aiming at obtaining the highest accuracy, most of the systems differ in so many dimensions that it is

[4] http://DBpedia-spotlight.github.io/demo/index.html
[5] http://demo.languagecomputer.com/cicerolite/
[6] http://www.zemanta.com/
[7] http://tagme.di.unipi.it/
[8] http://thewikimachine.fbk.eu/html/index.html
[9] http://en.wikipedia.org/wiki/Oak_Tree_(disambiguation)
[10] http://http://en.wikipedia.org/wiki/Major_Oak
[11] http://http://en.wikipedia.org/wiki/England
[12] Through this report we use the terms Entity Linking (EL) and Named Entity Linking (NEL) interchangeably.
[13] http://www.nist.gov/tac/2012/KBP/index.html

rather unclear which aspects of the systems are actually necessary for good performance and which aspects are harming it (Hachey et al., 2012).

The first large set of manually annotated named entity linking data was prepared for the KBP 2009 edition (McNamee et al., 2010). In the KBP 2012 edition, the reference KB is derived from English Wikipedia, while source documents come from a variety of languages, including English, Chinese, and Spanish. Other NEL evaluation datasets have been compiled independently of the KBP TAC shared tasks. These, together with some other resources based on Linked Data which can be used for NED will be listed and described in Section 2.2.

NED allows computing direct references to people, locations, organizations, etc. For example, in the financial domain NED can be used to link textual information about organizations to financial data, or in the tourist domain, NED can link information about hotels or destinations to particular opinions and/or facts about them.

Major components of a Named Entity Disambiguation System

There is a big heterogeneity regarding which techniques and algorithms NED systems currently use. However, almost all NED system can be classified as performing three major steps:

- Identification of mention strings: given a text document, the first step is to identify which text pieces are capable of being linked to external entities. For instance, given the example above, the first step would be to identify "Oak Tree" and "England" as entity mentions. Some systems (particularly, DBpedia Spotlight) use the term "spotting" to refer to the mention string identification step.
- Candidate selection: once entity mentions are identified, the next step is to generate the candidate entities for each particular mention string. Titles and other Wikipedia-derived aliases can be leveraged at this stage to capture synonyms. An ideal searcher should balance precision and recall to capture the correct entity while maintaining a small set of candidates. This reduces the computation required for disambiguation.
- Disambiguation: the last step involves choosing the best candidate among possible for each mention string.

2.2 Data Sources for NED

This section describes some of the relevant data sources for NED. The data sources are mainly either text corpora developed for NLP applications or Linked Data as part of the Linked Data[14] initiative. Most of the research on NED systems has been undertaken on text corpora, although some systems are already using Linked Data datasets such as DBpedia[15].

The data sources and systems described in this section will be those relevant to Wikification and NEL[16]. The datasets are not focused on any particular domain, and, particularly, they do not belong to the CH domain. However, these are the standard datasets used to evaluate and compare NED systems, and thus they are useful to draw conclusions on the benefits of each of the systems.

With the rise to prominence of Wikipedia, the Wikification task was proposed (Mihalcea and Csomai, 2007). Instead of clustering entities, as in Cross-document Coreference Resolution, mentions of important concepts in the text were to be linked to its corresponding Wikipedia article. Crucially, the Wikification task differs from (NEL) in that the concepts to be disambiguated are not necessarily named entities and in assuming that the knowledge base is complete.

As mentioned before, the first large datasets on NEL were created by the Text Analysis Conference (TAC) for the Knowledge Base Population (KBP) track. So far there have been 4 editions since 2009. Originally, the

[14] http://linkeddata.org/
[15] http://DBpedia.org/About
[16] The term NED will be used to refer to any of these two tasks indistinctly (Hachey et al., 2012)

datasets were only available for English but the 2012 edition includes documents in Spanish. In addition to the KBP datasets, several others have been created (Cucerzan, 2007; Fader et al., 2009). Furthermore, there is some work on integrating NEL annotation with existing NERC datasets such as the CONLL 2003 datasets (Hoffart et al., 2011).

Other valuable datasets listed in Table 1 for NED are those related with Linked Data. Linked Data is defined as "about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods". More specifically, Wikipedia defines Linked Data as "a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information and knowledge on the Semantic Web using URIs and RDF". Of course, the data to be linked can be any type of named entity currently available in the Web. Well-known and large linked data resources in the NLP community are DBpedia, Freebase[17] and Yago[18], but there are many others including those supported by large organizations such as the BBC, the British Government, NASA, CIA, Yahoo, etc. Current count in the list of Linked Data datasets is more than 300.

[17] http://www.freebase.com/

[18] http://www.mpi-inf.mpg.de/yago-naga/yago/

Data Entity	Type of data	Provision	Storage	Amount	Language	License	Website
KBP 2009	Newswire	Text corpora available from Linguistic Data Consortium (LDC)	Annotated files for development and evaluation	3904 instances	English	Private	http://apl.jhu.edu/~paulmac/kbp.html
KBP 2010	News, Blogs, Web data	Datasets available from LDC	Annotated files for development and evaluation	3750	English	Private	https://www.ldc.upenn.edu
KBP 2011	News, Web data	Datasets available from LDC	Annotated files for development and evaluation	~6000 instances for development, training and evaluation	English	Private	https://www.ldc.upenn.edu/
AIDA CoNLL YAGO	Newswire	Available as text corpora	Annotated files	34596 mentions	English	CC-BY-3.0 license, PU	http://www.mpi-inf.mpg.de/yago-naga/aida/downloads.html
Wikipedia Miner	Wikipedia, news	Text corpora	Annotated files	727 instances	English	Public	http://www.nzdl.org/wikification
DBpedia	Wikipedia articles	API, dump	Linked Data	~3.77 million named entities	English, Spanish, German, Dutch, Italian, French	CC-BY-SA license	http://DBpedia.org/

Table 1: Data Sources for Named Entity Disambiguation

KBP at TAC

The TAC KBP 2009 edition distributed a knowledge base extracted from a 2008 dump of Wikipedia and a test set of 3,904 queries. Each query consisted of an ID that identified a document within a set of Reuters news articles, a mention string that occurred at least once within that document, and a node ID within the knowledge base. Each knowledge base node contained the Wikipedia article title, Wikipedia article text, a predicted entity type (person, organization, location or misc), and a key-value list of information extracted from the article's infobox. Only articles with infoboxes that were predicted to correspond to a named entity were included in the knowledge base. The annotators favoured mentions that were likely to be ambiguous, in order to provide a more challenging evaluation. If the entity referred to did not occur in the knowledge base, it was labelled NIL. A high percentage of queries in the 2009 test set did not map to any nodes in the knowledge base: the gold standard answer for 2,229 of the 3,904 queries was NIL.

In the 2010 challenge the same configuration as in the 2009 challenge was used with the same knowledge base. In this edition, however, a training set of 1,500 queries was provided, with a test set of 2,250 queries. In the 2010 training set, only 28.4% of the queries were NIL, compared to the 57.1% in the 2009 test data and the 54.6% in the 2010 test data. This mismatch between the training and test data showed the importance of the NIL queries and it is argued that it may have harmed performance for some systems. This is because it can be quite difficult to determine whether a candidate that seems to weakly match the query should be discarded in favour of guessing a NIL. The most successful strategy to deal with this issue in the 2009 challenge was augmenting the knowledge base with extra articles from a recent Wikipedia dump. If a strong match against articles that did not have any corresponding node in the knowledge base was obtained, then NIL was return for these matches.

In the KBP 2012 edition, the reference KB is derived from English Wikipedia, while source documents come from a variety of languages, including English, Chinese, and Spanish.

AIDA CoNLL Yago

This corpus contains assignments of entities to the mentions of named entities annotated for the original CoNLL 2003 entity recognition task[19]. The entities are identified by YAGO2[20] entity name, by Wikipedia URL[21], or by Freebase[22]. The CoNLL 2003 dataset is required to create the corpus.

Wikipedia Miner

The Wikipedia Miner system was mainly tested on Wikipedia articles, by taking the links out and trying to put them back in automatically. In addition, the system was also tested on news stories from the AQUAINT corpus, to see if it would work as well "in the wild" as it did on Wikipedia. The stories were automatically wikified, and then inspected by human evaluators. This dataset contains the news stories of the AQUAINT corpus.

DBpedia

DBpedia is the Linked Data version of Wikipedia. The DBpedia data set currently provides information about more than 1.95 million "things", including at least 80,000 persons, 70,000 places, 35,000 music albums, 12,000 films classified in a consistent ontology. In total, it contains almost 4 million entities. It also provides descriptions in 12 different languages. Altogether, the DBpedia data set consists of (more than) 103 million RDF triples.

The data set is interlinked with many other data sources from various domains (life sciences, media, geographic government, publications, etc.), including the aforementioned Freebase and YAGO, among many others[23].

[19] http://www.cnts.ua.ac.be/conll2003/ner/

[20] http://www.mpi-inf.mpg.de/yago-naga/yago/

[21] http://en.wikipedia.org/wiki/Main_Page

[22] http://wiki.freebase.com/wiki/Machine_ID

[23] http://wiki.DBpedia.org/Datasets

2.3 Tools for NED

Most of the currently available systems have been developed as a result of the popularity of the Wikification and KBP tasks introduced in Section 2. Furthermore, the rise of Linked Data datasets have also contributed to the development of industrial NED systems. Most systems either perform Wikification (every concept is linked) or NEL (only named entities are disambiguated). Nowadays there are many NED systems for either Wikification or NEL. In this study, we focus on systems that comply to the follow requirements:

- Can be used as standalone programs or libraries. This options discards many systems which only provide web access.
- Have free licenses, so that systems can be distributed.

Table 2 lists the systems which fulfill the aforementioned requisites; thereafter some details of each system are provided.

System Service	Languages	Sources availability	Provision	Programming Language	License	Website URL
The Wiki Machine	English	Yes	Library			http://thewikimachine.fbk.eu/
Illinois Wikifier	English	Yes	Jar, Library	Java	Public	http://cogcomp.cs.illinois.edu/page/software_view/Wikifier
DBpedia Spolight	Dutch, English, German, Portuguese, Spanish	Yes	API, library, source code	Java	Apache 2.0, part of the code uses LingPipe Royalty Free license	http://DBpedia-spotlight.github.com/
WikiMiner	English, Spanish	Yes	Jar, library	Java	GNU GPLv3	http://wikipedia-miner.cms.waikato.ac.nz/
AIDA	English	Yes	Jar, Restful API	Java	CC-BY-NC-SA license	http://www.mpi-inf.mpg.de/yago-naga/aida/index.html

Table 2: Systems and Services for Named Entity Disambiguation

The Wiki Machine

The Wiki Machine is a Wikification system developed at the FBK in Trento, Italy. In addition to machine learning techniques, they use Linked Data to offer multilingual Wikification via DBpedia and Freebase. They also offer a publicly available demo which compares their results with respect to AlchemyAPI, Zemanta and OpenCalais.

Illinois Wikifier

The Illinois Wikifier system is developed at the Cognitive Computation Group at the University of Illinois at Urbana Champaign[24]. They present a Wikification system (Ratinov and Roth, 2009) using both local and global features. The results reported claim to outperform previous systems (Milne and Witten, 2008). It should be noted, however, that not many approaches to NED have evaluated their results with the same datasets, the KBP participants being the general exception. A newer version of the tool exists (Cheng and Roth, 2013).

DBpedia Spotlight

DBpedia Spotlight is a Wikification tool for automatically annotating mentions of DBpedia resources in text, providing a solution for linking unstructured information sources to the Linked Open Data cloud through DBpedia (Mendes et al., 2011; Daiber et al., 2013). DBpedia Spotlight recognizes that names of concepts or entities have been mentioned (e.g. "Michael Jordan"), and subsequently matches these names to unique identifiers (e.g. `DBpedia:Michael_I._Jordan`[25], the machine learning professor or `DBpedia:Michael_Jordan`[26] the basketball player).

DBpedia Spotlight can be used through their Web Application or Web Service endpoints. The Web Application is a user interface that allows entering text in a form and generates an HTML annotated version of the text with links to DBpedia. The Web Service endpoints provide programmatic access to the demo, allowing retrieval of data also in XML or JSON. DBpedia is released under the Apache License 2.0.

DBpedia Spotlight has two approaches. The original DBpedia Spotlight implementation uses Apache Lucene for disambiguation and LingPipe for spotting. Pre-built indexes and spotter models are available for English. The second approach is an statistical approach.

Wikipedia Miner

Wikipedia Miner is a Wikification system developed by the University of Waikato, New Zealand (Milne and Witten, 2008). The Wikipedia Miner can be used as a Web service or as a library via a Java API. The system uses machine learning and graph-based approaches to detect and disambiguate and link terms in running text to their Wikipedia articles. The system was the first publicly available tool for Wikification and many works still have it as a reference to evaluate their performance. Wikipedia Miner provided several benefits over previous Wikification work (Mihalcea and Csomai, 2007), by: (i) identifying in the input text of a set C of so-called *context pages*, namely, pages linked by spots that are not ambiguous because they only link to one article; (ii) calculating a *relatedness measure* between two articles based on the overlap between their in-linking pages in Wikipedia; and (iii) defining a notion of *coherence* with other context pages in the set C. These three main components of the system allowed them to obtain around 75% F measure over long and richly linked Wikipedia articles.

AIDA

AIDA is a framework and online tool for entity detection and disambiguation (Hoffart et al., 2011). Given a natural-language text or a Web table, it maps mentions of ambiguous names onto canonical entities (e.g., individual people or places) registered in the YAGO2 knowledge base[27].

[24] http://cogcomp.cs.illinois.edu/
[25] http://DBpedia.org/page/Michael_I._Jordan
[26] http://DBpedia.org/page/Michael_Jordan
[27] http://www.mpi-inf.mpg.de/yago-naga/yago/

Development		Test	
Dataset	Instances	Dataset	Instances
TAC 2010	1,675	TAC 2011	1,114
		AIDA	5,508

Table 3: Size of the datasets used in the evaluation

2.4 Evaluation

In the last section we describe the most relevant systems for state-of-the-art NED. An exhaustive evaluation of every system is clearly out of the scope this document, and therefore we will narrow our selection by refining the requirements presented in section 2.3 and putting additional requirements to the NED tools:

- The software is easy to download, build and install.
- The software is up-to-date and, if possible, there is an active community beyond it.
- The system is multilingual, or, at least, it is able to perform NED for English and Spanish.

Among the systems described in the previous section, only **DBpedia Spotlight** and **Wikipedia Miner** fulfill these additional requirements. Both are mature systems, are easy to build and install, and both systems have a community of users and developers willing to help in case of problems. We thus focus the evaluation on those systems only. To draw better conclusions from this evaluation exercise, we will be comparing the results obtained using the DBpedia Spotlight probabilistic models with those of the Wikipedia Miner. The evaluation consists of running both NED systems on a selection of the standard datasets described in section 2.2, assessing their overall performance, and testing different aspects such as memory, disk usage, etc. This section presents the results of this evaluation.

Both DBpedia Spotlight and Wikipedia Miner depend on a set of parameters which greatly affect the overall behavior of the systems. Given that we wanted to compare both systems in a fair way, we decided optimize the parameters of each system in one single dataset (the development dataset), and then try the best parameter combination on a different dataset (the test dataset). Table 3 shows the datasets used for development and testing.

The performance of the systems are measured using the standard **precision** and **recall** metrics. Precision is the number of correctly assigned instances divided by the total instances as returned by the system. Recall is the number of correctly assigned instances divided by the number of instances in the gold standard dataset. We also report **coverage** when required, the ratio of returned instances divided by the number of instances in the gold standard.

In our particular setting we seek to maximize precision, that is, we care more about returning correct links to DBpedia entities than trying to link all possible mentions in the input text. Because we focus our study on NED systems, we discard the so-called NIL instances (instances for which no correct entity exists in the Reference Knowledge Base) from the datasets.

2.4.1 DBpedia Spotlight

For this evaluation, we have used the so called statistical backend of DBpedia Spotlight[28], publicly available in their repository[29]. DBpedia spotlight has the following parameters:

[28] Available here: https://github.com/DBpedia-spotlight/DBpedia-spotlight/wiki/Statistical-implementation

[29] https://github.com/DBpedia-spotlight/DBpedia-spotlight/wiki/Statistical-implementation

Support	Confidence	Coref.	P	R
0	0	False	**89.23**	**73.92**
10	0.1	False	89.15	73.33
10	0.1	True	89.23	73.92

Table 4: Results of DBpedia Spotlight on the development dataset for different values for the Confidence and *Support* and Coref. parameters. P stands for precision and R stands for recall.

Mentions	Context	P	R
Entities	Window	89.23	**73.92**
Entities	Full	87.62	61.37
Any	Window	**90.02**	54.80
Any	Full	87.68	61.37

Table 5: DBpedia Spotlight results on the development dataset. The column Mentions describes whether all text is disambiguated ("Any") as opposed to evaluating just the named entities ("Entities"). The column Context distinguishes between using the whole document as context ("Full") or using a window of 50 words surrounding the target mention ("Window").

- Spotter: this parameter is used to specify which mentions should be disambiguated. Note that setting this parameter bypasses the spotter algorithm of Spotlight, as it assumes that another tool has performed spotting instead.
- CorefResolution: coreference resolution deals with the problem of identifying two or more expressions on a text which refer to the same entity. This parameter allows to activate/deactivate the coreference module within Spotlight. It should be noted that, despite the name given, this is not we we understand in NLP by "Coreference resolution". This module simply tries to say that *Michal Jordan* and *Jordan* are mentions of the same entity, but it does not deal with many other forms of nominal coreference resolution.
- Confidence: Disambiguate only when the selected entities have a confidence value above this parameter.
- Support: This is a disambiguation threshold. According to the documentation, Spotlight will disambiguate a mention only when the selected entities have a support value above this parameter. The documentation does not specify how the support value is calculated.

Table 4 shows the results for several parameter values of DBpedia Spotlight on the development dataset (TAC 2010). For these experiments we provided the tool with the mentions (spots), as we are mostly interested in analyzing the disambiguator performance for different parameters. The results show that DBpedia Spotlight disambiguator is very competitive and that it obtains performances that are very close to the best systems for this dataset. The table also shows that the system is robust with respect to the different values of parameters, and that there are no significant differences among them. Notably, coreference resolution module does not improve the overall performance.

With the table above we fixed the best parameters[30], and then we tried two additional settings:

- Whether to link any mention on the text the systems considers relevant or just the named entities.
- Whether to use the whole text or just a window of words around target mentions as disambiguation contexts.

[30] Support:0, Confidence:0, CorefResolution: False

		TAC 2011			AIDA		
Mentions	Context	P	R	mention/sec	P	R	mention/sec
Entities	Window	79.77	60.68	33	79.67	**75.94**	17
Any	Window	**87.31**	**61.83**	14	**81.94**	71.12	5

Table 6: DBpedia Spotlight results on the test dataset.

	TAC 2011			AIDA		
threshold	P	R	Coverage	P	R	Coverage
default	**87.31**	**61.83**	70.82	81.94	71.12	86.80
1.0	85.73	60.94	**71.09**	79.34	72.67	**91.59**

Table 7: Results on the test dataset for different values of spotter_threshold.

Table 5 shows the results on the development dataset. Using context instead of full document improves the results on this dataset. Note that the TAC challenge involves linking a single mention string from a possibly large document; the results suggest that using the whole document as context introduces too much noise into the system, probably coming from words that are very distant from the target mention string. The results also show that disambiguating only Named Entities leads to a higher recall, but lower precision.

With these results in hand, we decided to select the best setting[31] for the rest of the experiments. Table 6 shows the results of evaluating DBpedia Spotlight on the TAC 2011 and AIDA datasets using those parameters. The table also shows the mentions per second the tool is able to disambiguate on each dataset[32].

The results shows that DBpedia Spotlight achieves very good state-of-the-art results on these datasets. They also show that NED is a bit harder than Wikification, and that Wikification results obtain better overall performance although lower recall than NED.

We investigated further in order to increase the overall recall of the system by tweaking an additional parameter called spotter_threshold, which regulates the conditions a string has to meet to be considered as a mention. Table 7 shows the results on the test dataset for setting this parameter with the default value and at 1.0. The *coverage* column describes the percentage of mentions that are detected by the system. The results show that increasing the threshold does increase coverage, but at a loss of both precision and recall. In other words, the system is able to detect more mentions, but those new mentions are, in general, incorrectly disambiguated.

Spanish dataset: TAC 2012

DBpedia Spotlight provides pre-trained models for Spanish, extracted from the Spanish Wikipedia. Therefore, running Spotlight for Spanish is just a matter of choosing the Spanish model instead of the English one. We used the TAC 2012 Spanish dataset for testing Spotlight Spanish. Starting from 2012 the TAC/KBP conference includes a task on Cross-lingual Entity Linking for Spanish and Chinese. On this setting, systems are provided with a document in one language (Spanish or Chinese), and they have to link the mentions to entities belonging to an English Knowledge Base.

For evaluating the system we first run DBpedia Spotlight with Spanish over the TAC 2012 Spanish dataset, which outputs entities from Spanish DBpedia. We then map those entities to the corresponding English counterparts using the interlingual links from Wikipedia[33].

[31] Mentions: Any and Context: Window

[32] The experiments were performed on a server with one 2.1 GHZ processor and 64GB RAM.

[33] http://www.mediawiki.org/wiki/Interlanguage_links

Model	threshold	P	R	Cov
ES	default	78.15	55.80	71.40
ES	1.0	75.43	57.20	75.84
EN	default	**78.92**	58.40	74.00
EN	1.0	75.24	**59.26**	**78.76**

Table 8: Spotlight results on Spanish TAC2012 Dataset.

We tried the same set of parameters as used for the English experiments in the test dataset. Besides, we tried two additional settings:

- Whether to use the Spanish model or the English model.
- Change the spotter_threshold parameter to increase coverage.

Table 8 shows the results for this dataset. Surprisingly, the English model performs better than the Spanish model on this dataset. This results might be partially explained by the fact that the Spanish Wikipedia is smaller, so less context words might be available for each entity. The English wikipedia has more words and some of those words are entity names, which often have the same spelling across languages. In other words, it could be that overall, the English context is still richer than the Spanish.

All in all, the table shows that Spotlight lacks recall, that is, it is not able to correctly link all the mentions appearing in the document. However, when the system links a mention, it does so correctly over the 75% of the time. Increasing the spotter_threshold increases coverage and also yields to slightly better recall figures.

2.4.2 Wikipedia Miner

We used the 1.2.0 version of Wikipedia Miner trained with a dump from the 2011 English Wikipedia. Again, we used the development dataset to tune the parameters of the system. Wikipedia Miner has two main parameters:

- nimProbability: The system calculates a probability for each topic of whether a Wikipedian would consider it interesting enough to link to. This parameter specifies the minimum probability a topic must have before it will be linked.
- disambiguationPolicy: whether each term should be disambiguated to a single interpretation, or to multiple ones.

We tried different values of the parameters on the development dataset. Being a a Wikification tool, Wikipedia Miner always tries to disambiguate as many mentions in the text as possible. Therefore, unlike DBpedia Spotlight[34], we were not able to test the performance of Wikipedia Miner on a NED setting. In any case, the performance of Wikipedia Miner is much worse than DBpedia Spotlight: the best parameters yielded a precision of 57.07 and a recall of 63.43, a drop of 10 points in recall and almost 25 points in precision compared to the results of DBpedia Spotlight on the same dataset.

Table 9 confirms the previous results. The table shows the results of the best parameter combination for Wikipedia Miner when applied to the test dataset. Once again, Wikipedia Miner is outperformed by a large margin by DBpedia Spotlight in this dataset. We therefore discarded using Wikipedia Miner for the English background link micro-service.

[34] As described in the previous sections, DBpedia Spotlight is also a Wikification tool, but it is easily customizable by allowing to provide the Spotting by third party tools.

	P	R
TAC 2011	55.29	55.60
AIDA	74.80	58.49

Table 9: Wikipedia Miner results on the test dataset.

Regarding Spanish, we tried to run Wikipedia Miner on the TAC Spanish dataset with no success. Wikipedia Miner does not provide the Spanish models anymore (there are Spanish models available for previous versions of Wikipedia Miner, but not for the 1.2.0 version we were testing), so the only way to run Wikipedia Miner is by building the Spanish models again. Unfortunately, model building with Wikipedia Miner turned to be a very complex task full of subtle details and corner cases. In summary, we were not able to build the models and, thus, to properly test Wikipedia Miner on the Spanish dataset. However, Wikipedia Miner performs significantly worse than DBpedia Spotlight in the English dataset and we have no reasons to think that it would be different on the Spanish dataset.

2.5 Choosing a NED tool for the background link micro-service

Nowadays there are many tools which links pieces of text with appropriate entities belonging to a particular knowledge base such as Wikipedia or DBpedia. Although those systems are not focused on the CH domain, the fact that they link text to broad resources like Wikipedia or DBpedia make them appropriate for the background linking micro-service.

We have compared the results of two of the most widely used systems for NED, namely, DBpedia Spotlight and Wikipedia Miner, for both English and Spanish. The experiments clearly suggest using DBpedia Spotlight for the background link micro-service in both languages: DBpedia Spotlight performed better than Wikipedia Miner in our experiments, is easier to install and deploy on virtual machines and is fast enough to allow an interactive micro-service to be built on top of it.

DBpedia Spotlight have a set of parameters that tune its behavior. We have analyzed many variants of such parameters and measured the performance of the system on a development dataset. Overall, DBpedia Spotlight is very robust and have shown very slight variations on performance when changing those parameters.

As in many areas, there is a fundamental trade-off in NED between precision and recall, i.e., some systems try to disambiguate as much strings as possible whereas some others only do so when they are confident enough. For the purpose of this project, we have decided maximize precision at the expense of not disambiguating all potential strings of the text.

3 Vocabulary matching micro-service

Cultural Heritage institutions categorize their content by linking CH items with one or more relevant concepts from a controlled vocabulary. With the advent of information technology and the desire to make available CH resources to the general public, there is an increasing need to facilitate interoperability across these different contexts. There are many situations in which users may benefit from having the items organized into subject categories for browsing and exploration. For example, when users do not have clearly defined information needs (White et al., 2006), when attempting complex search tasks (Singer et al., 2012) or when they want to gain an overview over a collection (Hornbæk and Hertzum, 2011). In such cases the provision of only a simple search box is insufficient (Marchionini, 2006; Pirolli, 2009). This is particularly relevant to digital libraries where rich user/information interaction is common and requires alternative methods to support users (Rao et al., 1995). The provision of browsing functionalities through thesaurus-based search enhancements have all been shown to improve the search experience.

The aim of the vocabulary matching micro-service is to automatically assign relevant concepts and terms from selected vocabularies to CH items as provided by cultural institutions. The service receives metadata information as input, and return the items enriched with links to one or more relevant concept from a list of vocabularies.

The vocabulary matching micro-service requires a set of vocabularies to match the CH items against. Many schemes have been proposed to describe and manage cultural heritage data. Those schemes are usually found in form of classification schemes, subject heading lists, etc, and many times they are focused on a particular field, institution and even collections. Within the LoCloud project, Task 3.4 provides the *vocabulary service*[35], a collaborative platform to explore the potentials of crowdsourcing as a way of developing multilingual, semantic thesauri for local heritage content. In summary, the vocabularies used in the vocabulary matching micro-service are those developed by the community using the vocabulary service of Task 3.4.

The vocabulary matching micro-service consists of two modules. One module, called *vocabulary retriever*, retrieves the vocabularies from the vocabulary service and creates an internal database with the concepts and lexicalizations on several languages. This module is executed on a regular basis so that the local database is synchronized and up to date with the vocabularies from the vocabulary service, whose concepts and terms are updated in a continuous fashion. Once the vocabularies are retrieved and stored in the internal database, one second module, called the *matching module*, annotates CH items with appropriate concepts and terms as found in the vocabularies.

3.1 Retrieving the vocabularies

The vocabulary matching micro-service depends on a set of vocabularies, each one comprising a set of concepts and their lexicalizations on several languages. The *vocabulary retriever* module gathers all the content from vocabularies as found in the vocabulary server. The first step is to retrieve the list of vocabularies, as well as the URL of each vocabulary. Once the vocabulary list is retrieved from the Vocabulary server, each vocabulary is downloaded as a SKOS document format Miles et al. (2005). SKOS is an well known specification, built upon RDF and RDFs, for expressing the basic structure and content of concept schemes (thesauri, classification schemes, subject heading lists, taxonomies, terminologies, glossaries and other types of controlled vocabulary). It is published and maintained by the W3C Semantic Web Best Practices and Deployment Working Group. Figure 4 shows an excerpt of the "Genres" vocabulary following SKOS. The excerpt describes one concept of the vocabulary, whose preferred in English name is "Ballads" (`<skos:prefLabel>` element. The figure also shows the lexicalizations of the concept in German ("Balladen") and French ("ballades").

The SKOS vocabulary is parsed and the required information is extracted, namely, the vocabulary name, and a mapping between strings (the lexicalizations, which vary among languages) and the concepts they may refer to.

[35] Not to be confused with the vocabulary *matching* service, which is described here.

```
<skos:Concept rdf:about="http://test113.ait.co.at/tematres/vocab/?tema=501">
  <skos:prefLabel xml:lang="en">Ballads</skos:prefLabel>
  <skos:inScheme rdf:resource="http://test113.ait.co.at/tematres/vocab/"/>
  <skos:broader rdf:resource="http://test113.ait.co.at/tematres/vocab/?tema=1"/>
  <skos:broader rdf:resource="http://test113.ait.co.at/tematres/vocab/?tema=3"/>
  <skos:broader rdf:resource="http://test113.ait.co.at/tematres/vocab/?tema=2"/>
  <skos:broader rdf:resource="http://test113.ait.co.at/tematres/vocab/?tema=4"/>
  <skos:exactMatch>
    <skos:Concept rdf:about="http://test113.ait.co.at/tematres/dmGenresDE/?tema=15">
      <skos:prefLabel xml:lang="de">Balladen</skos:prefLabel>
    </skos:Concept>
  </skos:exactMatch>
  <skos:exactMatch>
    <skos:Concept rdf:about="http://test113.ait.co.at/tematres/dmGenresFR/?tema=15">
      <skos:prefLabel xml:lang="fr">ballades</skos:prefLabel>
    </skos:Concept>
  </skos:exactMatch>
  <dct:created>2013-12-03 11:42:15</dct:created>
</skos:Concept>
```

Figure 4: Excerpt of SKOS document for one terminology concept.

Note that relations among concepts, such as those described by the SKOS <skos:broader> relations, are not used. In other words, the vocabulary matching micro-service only requires a list of terms linked to the relevant concepts (a gazetteer). The parser would extract the following mappings from the excerpt showed in Figure 4. Note that mentions are lowercased, and white spaces are replaced by underscores.

String	Language	Concept
ballads	en	http://test113.ait.co.at/tematres/vocab/?tema=501
balladen	de	http://test113.ait.co.at/tematres/dmGenresDE/?tema=15
ballades	fr	http://test113.ait.co.at/tematres/dmGenresFR/?tema=15

Table 10 shows the list of vocabularies gathered by the retriever[36], including their sizes (number of concepts and terms) and supported languages. In total there are 29 vocabularies in three languages (English, German and French), which comprise circa $40,000$ concepts and $56,000$ different terms.

Synchronization issues

The vocabulary server implemented in Task 3.4 is meant to be a collaborative platform where users can continually create new terms and concepts or refine existing ones. The list of vocabularies, concepts and terms have thus a dynamic nature, and the vocabulary matching micro-service has to be synchronized with the vocabulary server to guarantee that the current version of the vocabularies is used.

The *vocabulary receiver* module is executed on a regular basis[37] and reconstructs an internal vocabulary database with the terms and concepts retrieved from the vocabulary server.

3.2 Matching CH items

The second module of the vocabulary matching micro-service is the so called *matching module*. This module receives CH items and matches the textual information with the appropriate terms from the vocabularies. The

[36] Note that this list may change as more and more vocabularies are integrated into the vocabulary server.

[37] At the time being the receiver is executed once per day, but the synchronization period can be changed easily.

matching module is the service exported to the users as a REST service, which can be called to enrich the CH items.

The service can be used to enrich any CH field, but the following fields are particularly relevant:

- The *description* field, a textual description of the CH item.
- The *subject* fields, so that the keywords described there can be linked to vocabulary terms.
- The *title* field.

The matching module analyzes the textual information and scans the tokens from left to right and consider the longest possible span which has a vocabulary entry as a candidate mention. Mentions are linked with the appropriate vocabulary terms for a particular language.

Vocabulary	Concepts	Labels	Lang
Alexandria Digital Library Feature Type Thesaurus	210	1285	en
Archeological Objects Thesaurus Scotland	213	263	en
Archeological Sciences Thesaurus	132	160	en
Building Materials Thesaurus	569	1009	en
Components Thesaurus	1435	1650	en
Event Type Thesaurus	115	158	en
Evidence Thesaurus	80	120	en
FISH Archeological Objects Thesaurus	2279	2865	en
General Multilingual Environmental Thesaurus GEMET	2011	2011	en
General Subject headings for Film Archives	1412	1987	en
Irish Monuments	1760	1760	en
Irish Periods	27	27	en
MDA Archaeological Objects Thesaurus	1649	2100	en
Maritime Craft Thesaurus Scotland	56	61	en
Maritime Craft Type Thesaurus	316	415	en
Monument Thesaurus Wales	1859	2571	en
Monument Type Thesaurus Scotland	1335	1705	en
Monument Type Thesaurus	6497	9580	en
Period Thesaurus Wales	15	16	en
Period Thesaurus	30	30	en
Relator Terms for Use in Rare Book and Special Collections Cataloguing	46	78	en
Tesauro de Ciencias de la Documentación	1593	1593	en
Thesaurus PICO 4.1	902	1038	en
Thesaurus for Graphic Materials 1: Subject Terms	2437	4508	en
Thesaurus for Graphic Materials 2: Genre and Physical Characteristic Terms	655	1198	en
UK Archival Thesaurus (UKAT)	1223	1715	en
UNESCO thesaurus	8605	12794	en,es
dm:Genres	1772	2688	fr,de,en
Total	39,233	55,385	en,fr,de,es

Table 10: Vocabularies, including the number of concepts and terms.

4 Implementation and examples of use

In this section we will briefly explain the implementation of both the background link and vocabulary matching micro-services. We also provide the actual Internet addresses for our entry points, as well as some examples of use.

4.1 Background link micro-service

The background link micro-service is built upon DBpedia Spotlight package, version 0.6, which is publicly available[38]. We used two instances of DBpedia Spotlight, one with the models trained for English and one with the models trained for Spanish for the English and Spanish background link micro-services, respectively[39]. Each DBpedia Spotlight instance is installed on a different virtual machine. The addresses of the virtual machines are the following:

URL	Service
http://test183.ait.co.at/rest/bglink	English language
http://lc013.ait.co.at/rest/bglink	Spanish language

Regarding parameters, we used the set of parameters that performed best in the datasets, as described in section 2.4.1. Those are the actual parameters:

- Spotter: Use Spotlight Spotter.
- CorefResolution: False.
- Confidence: 0.
- Support: 0.
- threshold: 1.0.

We implemented a PHP wrapper that acts as endpoint of the REST service. The wrapper listens to a particular port on the VM, reads the input as described in the API documentation, runs DBpedia Spotlight, and transforms the output to the required format, which is sent back to the caller.

Example of use

The following command sends the string "Guernica was painted by Picasso while he was in Paris." to the English background link micro-service[40].

```
curl -H "Accept: application/json"
-d "text=Guernica was painted by Picasso while he was in Paris."
http://test183.ait.co.at/rest/bglink
```

The response from the service is the following[41]:

[38] https://github.com/DBpedia-spotlight/DBpedia-spotlight/wiki/Statistical-implementation

[39] DBpedia Spotlight models are available here: http://spotlight.sztaki.hu/downloads/

[40] This example uses the *curl* application, which mimics a Web browser and can be used on a terminal. The same query can be made directly using any web browser, just by typing the address http://test183.ait.co.at/rest/bglink?text=Guernica%20was%20painted%20by%20Picasso%20while%20he%20was%20in%20Paris.

[41] The JSON format of the response is described in section 6.2

```
{"Status":200,
 "Status_message":"Success",
 "data":{
     "Resources":[
         {"URI":"http:\/\/DBpedia.org\/resource\/Guernica_(painting)",
          "similarityScore":"0.5724195306431602",
          "surfaceForm":"Guernica",
          "offset":"0"},
         {"URI":"http:\/\/DBpedia.org\/resource\/Pablo_Picasso",
          "similarityScore":"0.9996545655239845",
          "surfaceForm":"Picasso",
          "offset":"24"},
         {"URI":"http:\/\/DBpedia.org\/resource\/Paris",
          "similarityScore":"0.9998111742947978",
          "surfaceForm":"Paris",
          "offset":"48"}]}}
```

The background link micro-service has identified three entities and it has linked them to the appropriate DBpedia pages. The Spanish background link micro-service is very similar, the only difference being the actual address of the entry point. The following command sends the string "Picasso pintó el Guernica mientras vivía en París.":

```
  curl -H "Accept: application/json"
  -d "text=Picasso pintó el Guernica mientras vivía en París."
  http://lc013.ait.co.at/rest/bglink
```

which produces the following output:

```
{"Status":200,
 "Status_message":"Success",
 "data":{
     "Resources":[
         {"URI":"http:\/\/es.DBpedia.org\/resource\/Pablo_Picasso",
          "similarityScore":"0.9993425694476822",
          "surfaceForm":"Picasso",
          "offset":"0"},
         {"URI":"http:\/\/es.DBpedia.org\/resource\/Guernica_y_Luno",
          "similarityScore":"0.7089876977711708",
          "surfaceForm":"Guernica",
          "offset":"17"},
         {"URI":"http:\/\/es.DBpedia.org\/resource\/Par\u00eds",
          "similarityScore":"0.9999418831220671",
          "surfaceForm":"Par\u00eds",
          "offset":"44"}
     ]}}
```

5 Vocabulary matching micro-service

Section 3 explains the different modules implemented for the vocabulary matching micro-service, namely, the *vocabulary retriever* and the *matching module*. Both modules have been designed and implemented for LoCloud. Besides, we built another PHP wrapper which actually implements the REST service, much like in the background link micro-service. The address of the vocabulary matching micro-service is the following:

URL	Service
http://test183.ait.co.at/rest/vmatch	Vocabulary match

Example of use

The following command sends the text "Stembridge Windmill, High Ham, Somerset" to the vocabulary matching micro-service:

```
  curl -H "Accept: application/json"  \
  -d "text=Stembridge Windmill, High Ham, Somerset" \
  -d "lang=en" \
  http://test183.ait.co.at/rest/vmatch
```

The response is the following:

```
{"Status":200,
 "Status_message":"Success",
 "data":{
     "Resources":[
         {"URI":"http://test113.ait.co.at/tematres/Irish_Monuments/?tema=285",
          "vocab":"Irish Monuments"}
     ]}}
```

6 API reference

This section describes the complete API for both micro-services. We first describe the background link service API, and then the vocabulary matching API. Finally, we show the error codes for the the REST services, which are the same for both services.

6.1 Background Link micro-service

The background link micro-service receives raw UTF-8 text as input and produces the background link annotations. The output of the service is a JSON document with the annotated elements. This is its particular API:

- **endpoint**: The background link micro-service has two endpoints, depending on the language in which the CH items are described:
  - http://test183.ait.co.at/rest/bglink for the English background link service.
  - http://lc013.ait.co.at/rest/bglink for the Spanish background link service.
- **parameters**: just one parameter, "text", the text to annotate. The text has to be UTF-8 encoded.
- **example call**:

```
curl -H "Accept: application/json" \
-d "text=Guernica" \
http://test183.ait.co.at/rest/bglink
```

- **example output:**

```
{ "data" : {
  "Resources":
  [
    {"URI":"http://live.DBpedia.org/page/Guernica",
     "similarityScore":0.5,
     "surfaceForm":"Guernica",
     "offset":0},
    {"URI":"http://live.DBpedia.org/page/Guernica_(painting)",
     "similarityScore":0.5,
     "surfaceForm":"Guernica",
     "offset":0}
  ]}
}
```

- **Supported output types (POST/GET)**: application/json

The service produces a JSON document with an object of name `Resources`. The object contains a list of objects (one object per association), and each object of the list has the following elements:

- `URI`: the URI of a DBpedia page.
- `similarityScore`: a confidence value of the association.
- `surfaceForm`: the original text snippet from which the association was derived.
- `offset`: the starting position of the offset in the text. Useful if the original text contains many occurrences of the same mention string.

Request

Method	URL	Comment
POST	http://test183.ait.co.at/rest/bglink	English language
POST	http://lc013.ait.co.at/rest/bglink	Spanish language

Parameter	Datatype	Description
text	String	UTF-8 text to annotate.

Response

Status	Response
200	JSON record with the matched URIs. For example: {"URI":"http://live.DBpedia.org/page/Guernica", "similarityScore":0.5, "surfaceForm":"Guernica", "offset":0}, {"URI":"http://live.DBpedia.org/page/Guernica_(painting)", "similarityScore":0.5, "surfaceForm":"Guernica", "offset":0}

Parameter	Description
URI	URI of a DBpedia page.
similarityScore	Confidence value of the association.
surfaceForm	Original text snippet from which the association was derived.
offset	Starting position of the offset in the text.

6.2 Vocabulary Matching micro-service

The vocabulary link micro-service receives raw UTF-8 text as input and produces the annotations to vocabulary concepts. The output of the service is a JSON document with the annotated elements. This is its API specification:

- **endpoint**: The endpoint for this service is http://test183.ait.co.at/rest/vmatch
- **parameters**:
  - **text**: the text to annotate. The text has to be UTF-8 encoded.
  - **lang**: a language tag following the `xml:lang` format. If omitted, the default "en" value will be used.
- **example call**:

```
curl -H "Accept: application/json"  \
-d "text=Stembridge Windmill, High Ham, Somerset" \
-d "lang=en" \
http://test183.ait.co.at/rest/vmatch
```

- **example output:**

```
{ "data" : {
  "Resources":
  [
         {"URI":"http://test113.ait.co.at/tematres/Irish_Monuments/?tema=285",
          "vocab":"Irish Monuments"}
  ]}
}
```

- **Supported output types (POST/GET)**: application/json

The service produces a JSON document with an object of name `Resources`. The object contains a list of objects (one object per association), and each object of the list has the following elements:

- `URI`: the URI of a vocabulary concept.
- `vocab`: the name of the vocabulary used for matching.

Request

Method	URL
POST	http://test183.ait.co.at/rest/vmatch

Parameter	Datatype	Description
`text`	String	UTF-8 text to annotate.
`lang`	String	Two letter language code.

Response

Status	Response

200	JSON record with the matched URIs. For example:
	`{"URI":"http://test113.ait.co.at/tematres/Irish_Monuments/?tema=285",`
	`"vocab":"Irish Monuments"}`

Parameter	Description
`URI`	URI of the vocabulary concept.
`vocab`	Vocabulary name.

6.3 HTML Status Codes

All status codes are standard HTTP status codes. The below ones are used in both APIs.

Status Code	Description
200	OK
201	Created
202	Accepted (request accepted and queued for execution)
400	Bad request
401	Authentification failure
403	Forbidden
404	Resource not found
405	Method not allowed
409	Conflict
412	Precondition failed
413	Request entity too large
500	Internal server error
501	Not implemented
503	Service unavailable

7 Conclusion

This deliverable describes the metadata enrichment services as implemented and deployed in the LoCLoud project. The enrichment services comprises the *background link* micro-service (for English and Spanish), which links CH items to DBpedia elements, and the *vocabulary matching* micro-service, which maps CH items with relevant vocabulary concepts.

The background link micro-service relies on DBpedia Spotlight, a state-of-the-art tool for performing Named Entity Disambiguation (NED). DBpedia Spotlight was chosen because it performed best in our experiments, as explained in the deliverable.

The vocabulary matching micro-service is developed from scratch for the project. It synchronizes with the *vocabulary service*[42], a collaborative platform to explore the potentials of crowdsourcing as a way of developing multilingual, semantic thesauri for local heritage content.

All micro-services are implemented as REST services and are deployed into virtual machines (VM). Documentation on the service, including full API specification, is avaliable at this address:

http://support.locloud.eu/LoCloud%20Enrichment%20Microservice

The metadata enrichment micro-services are used in the various enrichment workflows automatically through the LoCloud Generic Enrichment Service, which allows to orchestrate various REST micro-services into complex enrichment workflows.

[42] Task 3.4 of LoCloud project.

References

Eneko Agirre and Philip Edmonds. *Word Sense Disambiguation: Algorithms and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2007. ISBN 1402068700, 9781402068706.

Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 79–85, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, 2006. URL http://acl.ldc.upenn.edu/E/E06/E06-1002.pdf.

Xiao Cheng and Dan Roth. Relational inference for wikification. In *EMNLP*, 2013.

S. Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of EMNLP-CoNLL*, volume June, pages 708–716, 2007. URL http://acl.ldc.upenn.edu/D/D07/D07-1074.pdf.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 121–124, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1972-0. doi: 10.1145/2506182.2506198. URL http://doi.acm.org/10.1145/2506182.2506198.

Martin Doerr, Stefan Gradmann, Steffen Hennicke, Antoine Isaac, Carlo Meghini, and Herbert van de Sompel. The europeana data model (EDM). In *World Library and Information Congress: 76th IFLA general conference and assembly*, pages 10–15, 2010.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Scaling wikipedia-based named entity disambiguation to arbitrary web text. In *Proceedings of {WIKIAI09}*, 2009.

B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J.R. Curran. Evaluating Entity Linking with Wikipedia. *Artif. Intell.*, 194:130–150, January 2012. ISSN 0004-3702. doi: 10.1016/j.artint.2012.04.005. URL http://dx.doi.org/10.1016/j.artint.2012.04.005.

B. Haslhofer, E. M. Roochi, M. Gay, and R. Simon. Augmenting europeana content with linked data resources. Proceedings of the 6th International Conference on Semantic Systems, pages 40:1–40:3, New York, NY, USA, 2010.

J. Hoffart, M.A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust Disambiguation of Named Entities in Text. In *Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, United Kingdom 2011*, pages 782–792, 2011.

Kasper Hornbæk and Morten Hertzum. The notion of overview in information visualization. *International Journal of Human-Computer Studies*, 69(7-8):509 – 525, 2011. ISSN 1071-5819. doi: 10.1016/j.ijhcs.2011.02.007. URL http://www.sciencedirect.com/science/article/B6WGR-529V18J-1/2/95a091a9a1a8d5423cd3fbdbd6ff5fc2.

Gary Marchionini. Exploratory search: From finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

P. McNamee, H.T. Dang, H. Simpson, P. Schone, and S.M. Strassel. An Evaluation of Technologies for Knowledge Base Population. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, page 369–372., 2010.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0621-8.

Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.

Alistair Miles, Brian Matthews, Michael Wilson, and Dan Brickley. Skos core: Simple knowledge organisation for the web. In *Proceedings of the 2005 International Conference on Dublin Core and Metadata Applications: Vocabularies in Practice*, DCMI '05, pages 1:1–1:9. Dublin Core Metadata Initiative, 2005. ISBN 8489315442, 9788489315440. URL http://dl.acm.org/citation.cfm?id=1383465.1383467.

D. Milne and I.H. Witten. Learning to Link with Wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*, page 509, New York, New York, USA, 2008. ACM Press. ISBN 9781595939913.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007.

Peter Pirolli. Powers of 10: Modeling complex information-seeking systems at multiple scales. *Computer*, 42(3): 33–40, 2009. doi: 10.1109/MC.2009.94.

Ramana Rao, Jan O. Pedersen, Marti A. Hearst, Jock D. Mackinlay, Stuart K. Card, Larry Masinter, Per-Kristian Halvorsen, and George C. Robertson. Rich interaction in the digital library. *Commun. ACM*, 38(4):29–39, April 1995. ISSN 0001-0782. doi: 10.1145/205323.205326. URL http://doi.acm.org/10.1145/205323.205326.

Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-29-9.

G. Singer, U. Norbisrath, and D. Lewandowski. Ordinary search engine users carrying out complex search tasks. *Journal of Information Science*, 2012.

Ryen W. White, Bill Kules, Steven M. Drucker, and M.C. Schraefel. Introduction. *Commun. ACM*, 49(4):36–39, April 2006. ISSN 0001-0782. doi: 10.1145/1121949.1121978. URL http://doi.acm.org/10.1145/1121949.1121978.