Deliverable D2.3 Specification of Web mining process for hypervideo concept identification

Ivo Lašek, Tomáš Kliegr, Milan Dojchinovski / UEP
Mathilde Sahuguet, Giuseppe Rizzo, Benoit Huet, Raphaël Troncy / EURECOM

08/10/2012

Work Package 2: Linking hypervideo to Web content

LinkedTV
Television Linked To The Web
Integrated Project (IP)
FP7-ICT-2011-7. Information and Communication Technologies
Grant Agreement Number 287911

Dissemination level	PU
Contractual date of delivery	30/09/2012
Actual date of delivery	08/10/2012
Deliverable number	D2.3
Deliverable name	Specification of Web mining process for hypervideo concept identification
File	`LinkedTV_D2.3_Specification_Of_Web_Mining_Process.tex`
Nature	Report
Status & version	Final & v1.0
Number of pages	64
WP contributing to the deliverable	2
Task responsible	UEP
Other contributors	EURECOM
Author(s)	Ivo Lašek, Tomáš Kliegr, Milan Dojchinovski / UEP Mathilde Sahuguet, Giuseppe Rizzo, Benoit Huet, Raphaël Troncy / EURECOM
Reviewer	Dorothea Tsatsou / CERTH
EC Project Officer	Thomas Kuepper
Keywords	Named Entity Recognition, NERD, Web mining, Multimedia retrieval
Abstract (for dissemination)	This deliverable presents a state-of-art and requirements analysis report for the web mining process as part of the WP2 of the LinkedTV project. The deliverable is divided into two subject areas: a) Named Entity Recognition (NER) and b) retrieval of additional content. The introduction gives an outline of the workflow of the work package, with a subsection devoted to relations with other work packages. The state-of-art review is focused on prospective techniques for LinkedTV. In the NER domain, the main focus is on knowledge-based approaches, which facilitate disambiguation of identified entities using linked open data. As part of the NER requirement analysis, the first tools developed are described and evaluated (NERD, SemiTags and THD). The area of linked additional content is broader and requires a more thorough analysis. A balanced overview of techniques for dealing with the various knowledge sources (semantic web resources, web APIs and completely unstructured resources from a white list of web sites) is presented. The requirements analysis comes out of the RBB and Sound and Vision LinkedTV scenarios.

History

Table 1: History of the document

Date	Version	Name	Comment
24/07/2012	v0.1	Kliegr, UEP	first version of the deliverable
24/07/2012	v0.2	Troncy, EURECOM	revised version of the TOC
07/08/2012	v0.31	Lasek, UEP	Knowledge Based Approaches to NER, SemiTags evaluation
08/08/2012	v0.32	Kliegr, UEP	first version for 1.4.1, 3.2.4, 3.3.3, 4.1
08/08/2012	v0.33	Lasek, UEP	Retrieval from Structured Sources, NER Web APIs
09/08/2012	v0.34	Lasek, UEP	Functional Requirements
10/08/2012	v0.35	Lasek, UEP	Crawling from Unstructured Sources
14/08/2012	v0.36	Lasek, UEP	Wrappers
15/08/2012	v0.37	Lasek, UEP	Retrieval from Public APIs
17/08/2012	v0.4	Lasek, UEP	State of the Art in Named Entity Recognition
23/08/2012	v0.5	Kliegr, UEP	Revision of UEP contribution, Introduction, Conclusion
27/08/2012	v0.6	Lasek, UEP	Retrieving Content by Text Analysis, SemiTags
29/08/2012	v0.61	Sahuguet, EURECOM	Workflow, Public APIs, section 3 and 5
03/09/2012	v0.7	Lasek, UEP	Completed sections Structural and Textual Disambiguation
03/09/2012	v0.71	Kliegr, UEP	Section 3: Renamed subsections and expanded intro
07/09/2012	v0.8	Sahuguet, EURECOM	Re-structure the deliverable after QA from Lyndon
19/09/2012	v0.81	Sahuguet, EURECOM	Requirements from scenarios and reworked the introduction
20/09/2012	v0.82	Troncy, EURECOM	Re-work entire sections 2 and 4
02/10/2012	v0.9	Kliegr, UEP	emphasize focus on text analysis
02/10/2012	v0.91	Kliegr, UEP	add section 5.4.3 supporting the requirements
08/10/2012	v0.92	Troncy, EURECOM	add NERD evaluation and full proof read of the deliverable
11/10/2012	v1.0	Troncy, EURECOM	Martha's edits and last comments addressed

0 Table of Content

1 Introduction

This deliverable provides a review of the state-of-the-art in the areas of Web mining for hypervideo concept identification. The overall goal is to analyze textual resources and metadata provided by multimedia analysis associated with a LinkedTV seed video in order to provide either structural information or related multimedia content that could be used for enriching the seed video. This additional content is provided to WP4 which aims to apply a personalization layer on top of these suggestions, and to WP3 which will practically display the additional information in the rich hypervideo LinkedTV player.

The starting point of this work are the results of the analysis performed by WP1 on seed videos together with other metadata (legacy, subtitles) provided by content providers. The first step of our approach is to extract named entities, associate them types or categories and disambiguate them with unique identifiers which are generally linked data resources. This will complement the annotations performed by WP1 and will generate new annotated video fragments. The second step aims at providing additional content in relation with the seed video fragments. Annotations available in the LinkedTV platform can be the source of queries for retrieval of this additional content (e.g. directly from existing structured LOD datasets or generated by mining white lists of unstructured web sites). For most of the video fragments, we aim at proposing a set of hyperlinked media that will further be filtered by WP4 and presented by WP3, in order to display relevant additional content to the user according to his/her profile. The purpose of this deliverable is to provide other partners, particularly those in WP3 (Presentation), WP4 (Personalization) and WP6 (Scenarios), in the project with a concise description of Web mining techniques, tasks and the intended processes suitable for LinkedTV.

Named entity identification and retrieval of additional content are carried out within WP2. Figure 1 illustrates the relationship of WP2 to other work packages of LinkedTV. Analysis of the seed video being watched by a LinkedTV user is made on the basis of metadata, primarily the work generated by WP1 and some additional data such as the potential legacy metadata provided by a broadcaster. Other inputs to the work package are those needed for retrieval of additional content including curated or white-list of web resources. The results of the Web mining process are both (named) entities found in the seed

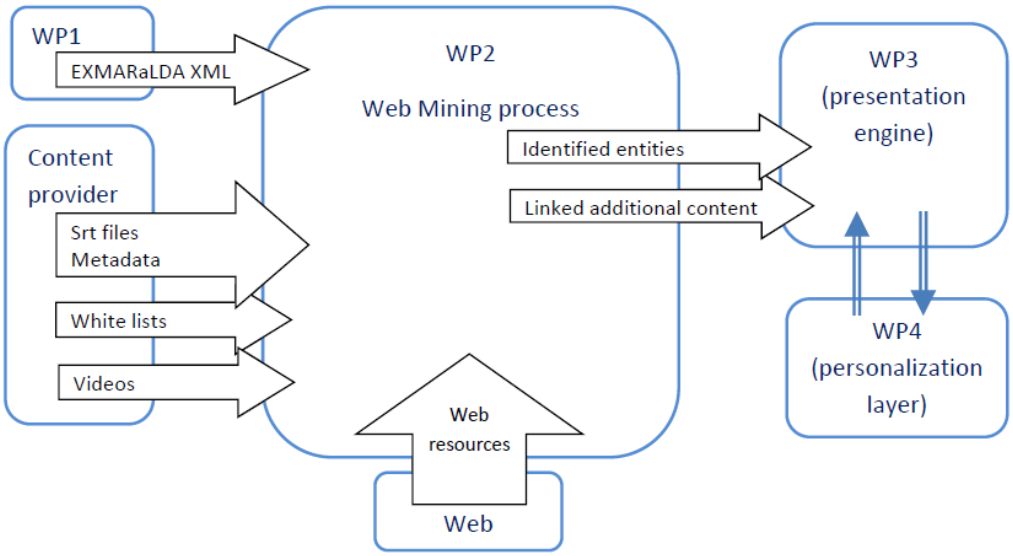


Figure 1: Inputs and Outputs of WP2

content and uniquely disambiguated with URIs, and additional content related to the seed video. They form a content pool that is candidate to be displayed to the LinkedTV user via the LinkedTV presentation engine (WP3), subject to filtering, approval and personalization (WP4) when appropriate. The result of this process will ultimately be rendered by the LinkedTV media player.

The work carried out within WP2 is structured around two axes (Figure 2):

– The conversion of both the legacy metadata and the results of the automatic multimedia analysis performed in WP1 and serialized in the eXmaralda file into RDF triples that are stored into a triple

store (hosted by Condat). Named entity recognition is made in parallel on either the transcripts provided by the broadcaster or the automatic ASR performed by WP1. The named entities extracted are themselves used in additional RDF annotations.

– The mining and retrieval part which aims at linking additional content to the seed videos being watched by the LinkedTV user. This additional content is looked up from a curated or white list of web site using different methods as explained in Figure 2.

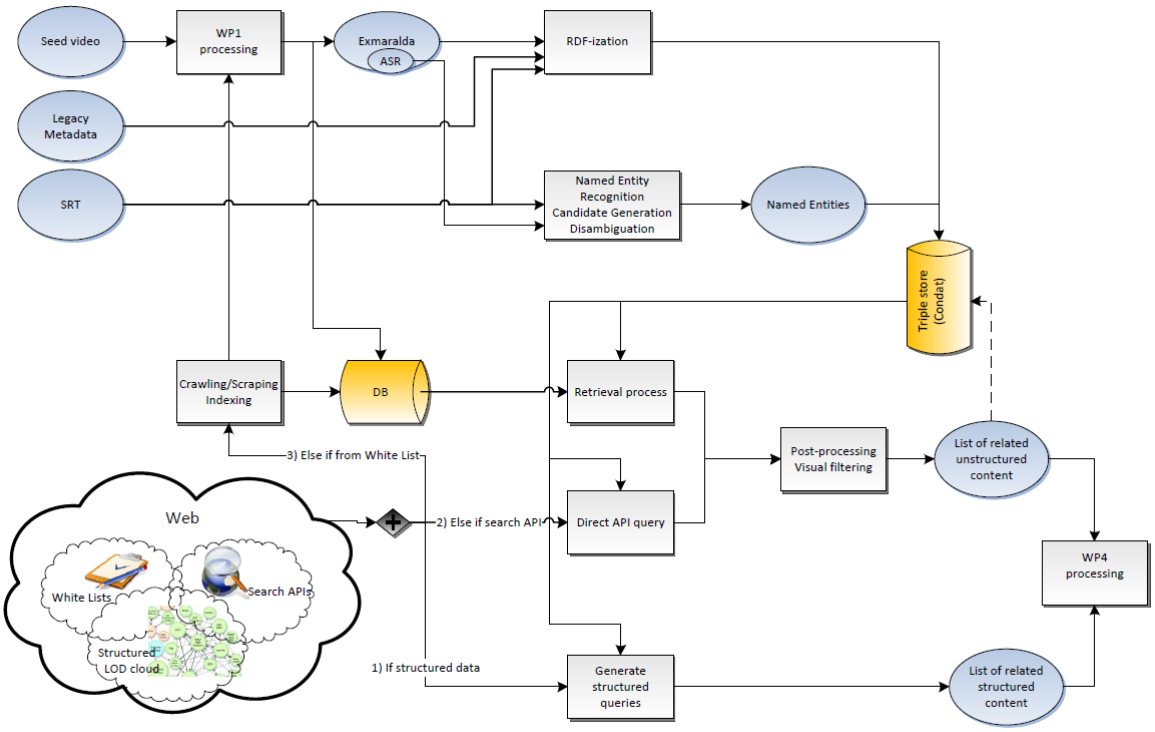


Figure 2: WP2 general workflow

The review of the state-of-the art is divided into two sections. Section 2 reviews techniques for named entity recognition and disambiguation. The three types of techniques covered are statistical approaches, knowledge-based approaches and Web APIs. Section 3 reviews the area of retrieval of additional content from the Web. This section describes crawling of structured and unstructured resources as well as public APIs.

The requirements, specification and results of the tools developed so far for entity recognition and additional content retrieval are presented in Section 4 and Section 5. Subsection 4.2 presents NERD, a complete framework for performing named entity recognition and disambiguation. We also describe the two novel tools named SemiTags and Targeted Hypernym Discovery (THD) which can perform Named Entity Recognition (NER) on the German and Dutch languages. In particular, SemiTags is a Web service interface to two existing statistical algorithms which were trained for Dutch and German while THD is a knowledge-based (Wikipedia-based) algorithm developed within the consortium. Section 5 describes the requirements and specification for retrieval of additional content from the Web. The work in this area is just starting, therefore no software results are yet presented. Finally, we conclude this deliverable and outline future work in the Section 6.

2 State of the Art in Named Entity Recognition and Disambiguation

Originally, Named Entity Recognition (NER) is an information extraction task that seeks to locate atomic elements in text. The NER and disambiguation problems have been addressed in different research fields such as NLP, Web mining and Semantic Web communities. All of them agree on the definition of a Named Entity, which was coined by Grishman et al. as an information unit described by the name of a person or an organization, a location, a brand, a product, a numeric expression including time, date, money and percent found in a sentence [GS96].

Initially, these NER techniques focused on identifying atomic information unit in a text (the named entities), later on classified into predefined categories (also called context types) by classification techniques, and linked to real world objects using web identifiers. Such a task is called Named Entity Disambiguation. The NER task is strongly dependent on the knowledge base used to train the NE extraction algorithm. Leveraging on the use of DBpedia, Freebase and YAGO, recent methods coming from the Semantic Web community have been introduced to map entities to relational facts exploiting these fine-grained ontologies. In addition to detect a Named Entity (NE) and its type, efforts have been spent to develop methods for disambiguating information unit with a URI. Disambiguation is one of the key challenges in this scenario and its foundation stands on the fact that terms taken in isolation are naturally ambiguous. Hence, a text containing the term London may refer to the city London in UK or to the city London in Minnesota, USA, depending on the surrounding context. Similarly, people, organizations and companies can have multiple names and nicknames. These methods generally try to find in the surrounding text some clues for contextualizing the ambiguous term and refine its intended meaning. Therefore, a NE extraction workflow consists in analyzing some input content for detecting named entities, assigning them a type weighted by a confidence score and by providing a list of URIs for disambiguation. The problem of word sense disambiguation is defined as the task of automatically assigning the most appropriate meaning to a polysemous word within a given context. Such a word sense disambiguation facilitates more accurate information filtering and enables enhanced text browsing. In multimedia context, named entity recognition helps in retrieval of additional related content and locating related videos [BCD05].

The named entity recognition and disambiguation process consists generally in the following steps:

- Named Entity Recognition – Identification of named entities in a given text.
- Candidate Generation – Finding possible word senses or identifiers of concrete candidate entities that can occur under the recognized surface form.
- Disambiguation – Selecting the most appropriate meaning (concrete category or identifier from a knowledge base) within a given context.

In the following sections, we describe the two main approaches for performing named entity recognition:

- Statistical approaches grounded in computational linguistics that often use some representation of an entity context to classify it in a predefined or open set of categories (section 2.1).
- Knowledge based approaches that aim at mapping recognized entities to concrete records in a backing knowledge base[1] (section 2.2). An advantage of such a detailed disambiguation is the possibility to enrich unstructured text with additional structured data from the knowledge base beyond just the type of an entity.

We conclude this section by describing web APIs that offer named entities and disambiguation functionalities (section 2.3) and a comparison of those APIs (section 2.4).

2.1 Statistical Approaches Grounded in Computational Linguistics

Early studies were mostly based on hand crafted rules, but most recent ones use supervised machine learning as a way to automatically induce rule-based systems or sequence labeling algorithms starting from a collection of training examples. However, when training examples are not available, even recent approaches stick with some kind of hand crafted rules often backed by a knowledge base [SNN04]. Statistical approaches to named entity recognition can be divided into three groups: Supervised Learning Approaches, Semi-Supervised Learning Approaches and Unsupervised Learning Approaches.

[1] Such a knowledge base can include a proprietary data source like social networks for names of people or a general data source such as Wikipedia or DBpedia.

2.1.1 Supervised Learning

The idea of supervised learning is to study the features of positive and negative examples of named entities over a large collection of annotated documents and design (learn) rules that capture instances of a given type. Supervised machine learning techniques include Hidden Markov Models [BMSW97], Decision Trees [Sek98], Maximum Entropy Models [BSAG98], Support Vector Machines [AM03] and Conditional Random Fields [LMP01, ML03, FGM05].

In [NL96], the LEXAS system is described as using a wide range of features that can be used to train the disambiguation algorithm. These include Part of Speech (POS) tags of surrounding words, POS tag of the disambiguated word, surrounding words in their basic form, collocations (words or phrases often co-occurring with the given sense), verb-object syntactic relations. LEXAS determines the correct meaning of the word by looking for the nearest meaning in terms of the features. In [Ped01], bigrams occurring nearby the disambiguated word are used as features. Weka [WF99] implementations of the C4.5 decision tree learner, the decision stump and the Naive Bayesian classifier are used.

2.1.2 Semi-Supervised Learning

As opposed to supervised learning methods, semi-supervised methods require only a limited set of examples or initial seeds in order to start the learning process. For example, the system may ask for a limited number of names of sought entities. They are then located in a text and the system tries to identify some contextual features characteristic for all the located entities. The results are then used to locate additional entities found in similar contexts. The learning process is then repeated.

In [NTM06] a named entity extractor exploits the HTML markup of Web pages in order to locate named entities. It is reported to outperform baseline supervised approaches but it is still not competitive with more complex supervised systems.

In [Bri99] semi-supervised learning is used to extract names of books and their authors. At the beginning example pairs of `author name - book name` are given. They are used to learn patterns that model the context of these pairs. A limited class of regular expressions is used for the patterns. Such derived patterns are then used to extract new names.

Collins and Singer [CSS99] use unlabeled data directly through co-training. They rely upon POS-tagging and parsing to identify training examples and patterns. Patterns are kept in pairs $spelling, context$ where spelling refers to the proper name and context refers to the noun phrase in its neighborhood. The training starts with a limited group of spelling rules. They are used to identify candidates in a text and classify them. The most frequent candidate contexts are used to derive contextual rules which can in turn be used to identify further spelling rules.

In [RJ99], the algorithm starts with a set of seed entity examples of a given type. At the heart of the approach, there is a mutual bootstrapping technique that learns extraction patterns from the seed words and then exploits the learned extraction patterns to identify more words that belong to the semantic category. More fine-grained context representation is introduced in [CV01] where elementary syntactic relations [BPV94] are used.

A Web scale fact extraction is performed in [PLB+06]. The recall of fact extraction is increased by pattern generalization - words from the same class are replaced by the same placeholder. The authors report a precision of about 88% by 1 million extracted facts from 100 million Web documents.

Ensembles are used in [PP09]. Combination of distributional [PLB+06] and pattern-based [PCB+09] algorithms is re-implemented. A gradient boosted decision tree is used to learn a regression function over the feature space for ranking the candidate entities. Another example of Web scale named entity recognition is given in [WKPU08]. A wide variety of entity types is recognized. Training data is automatically generated from lists on Web pages (tables and enumerations) and again by deriving patterns (templates). However, templates are used as a filter, rather than as an extraction mechanism.

In [GGS05] a similar task of word sense disambiguation is supported by semantic resources obtained from large corpora where terms are mapped to domains. This domain model is constructed in the completely unsupervised way using clustering based on Latent Semantic Analysis. The authors report that such a domain model contributes to better results even with limited amount of training data that are often difficult to gather.

2.1.3 Unsupervised Learning

An example of unsupervised named entity recognition using WordNet is given in [AM02]. The aim is to assign a known concept from WordNet to an unknown concept in a text. It is achieved by analysing

words that often co-occur with each known concept. Certain language patterns (e.g. such as, like, or other) are exploited in [Eva03]. The Google search engine is used to locate additional hypernyms. The sets of hypernyms are then clustered in an attempt to find general types of named entities. An observation that a Named Entity is likely to appear synchronously in several news articles, whereas a common noun is less likely is exploited in [SS04]. Authors report they successfully obtained rare Named Entities with 90% accuracy just by comparing time series distributions of a word in two newspapers.

KnowItAll [Etz05] uses the redundancy of the Web to perform a bootstrapped information extraction process. As one of the features that serve as an input for Naïve Bayesian Classifier a pointwise mutual information (PMI) [Tur01] is used. The PMI is counted between each extracted instance and multiple, automatically generated discriminator phrases associated with the class.

2.1.4 Summary

Statistical-based approaches often do not disambiguate entities into many diverse categories. Hence, the standard types used are: people, locations, organizations and others. From this point of view, knowledge-based approaches are more suitable for the need of LinkedTV: finding unique identifiers that disambiguate named entities and obtaining additional information for these named entities. However, statistical approaches provide very good results in the process of named entity recognition in texts. The de facto standard state-of-the-art solution in this area is the Stanford Named Entity Recognizer [FGM05] which exploits conditional random fields (CRF) models [ML03].

CRF belongs to the group of supervised learning algorithms and, as such, needs a comprehensive training data set. This could be an issue since LinkedTV has to deal with at least three different languages (English, German and Dutch). The authors provide models for English texts. Trained models for German can be found in [FP10]. Fortunately, the CoNLL 2003 shared task[2] [TKSDM03] provides a comprehensive annotated corpus for various languages including Dutch.

Additional semi-supervised and unsupervised techniques can be used in later stages of the project in order to improve the named entity recognition process. The approaches to extraction of further information about named entities [Bri99] and exploiting HTML structure of web pages [NTM06] can be used to enhance indexing and retrieval of additional content. This is the subject of our further evaluation.

2.2 Knowledge Based Approaches

Apart from statistical approaches to named entity recognition, the recognition and disambiguation may be supported by a knowledge base. The knowledge base serves on one hand as the white list of names that are located in a text. On the other hand, many services supported by a knowledge base assign concrete identifiers to recognized entities and thus can be mapped to additional information describing the recognized entities. Many general purpose named entity recognition tools use DBpedia [BLK+09] as their knowledge base (e.g. DBpedia Spotlight [MJGSB11a], Wikify [MC07]) or map recognized named entities directly to Wikipedia articles [BPP06].

Sometimes, limiting the recognition to only a constrained domain may improve the results for domain specific application. In [GNP+09], the authors deal with texts written in informal English by restricting the named entity recognition to the music domain. MusicBrainz [Swa02] is used as a backing knowledge base. In [HAMA06], the authors use a specific ontology for person names disambiguation. They disambiguate names of researchers in posts from DBWorld [dbw06], using DBLP [Ley02] as a knowledge base. Person names disambiguation is examined also in [Row09]. Here, a social graph mined from social networks is used as a knowledge base. An example of named entity recognition in the geospatial domain is given in [VKMM07] that uses data from GNIS [GNI12] and Geonet [geo12] combined with Wordnet [Mil95] as a knowledge base.

One of the most popular knowledge bases remains Wikipedia. It was used also in [MC07, MJGSB11a, MW08a]. A big advantage of Wikipedia is that links created in articles by Wikipedia contributors can be used as manual annotations. Each link to a Wikipedia article represents a mention of an entity represented by the target article. In Figure 3, we show an example of links in a Wikipedia article and the representation of their anchor texts in the source of this article. We can see that the entity British Empire[3] has the same anchor text, whereas the entity American Revolutionary War[4] has the anchor text American Revolution, which is an alternative surface form for this entity.

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

[3] http://en.wikipedia.org/wiki/British_Empire

[4] http://en.wikipedia.org/wiki/American_Revolutionary_War

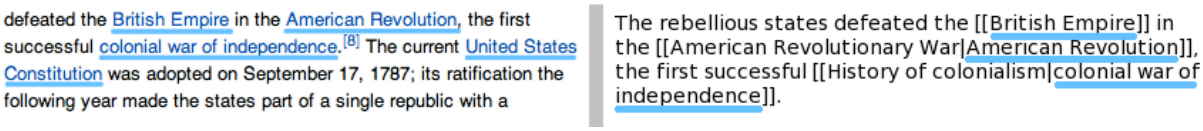


Figure 3: A sample of links in a Wikipedia article together with their representation in the source of a Wikipedia article.

One important feature of an entity is its commonness [MWM08] (i.e. prior probability of a particular sense of a given surface form). In the case of Wikipedia, this is usually measured as the count of incoming links having a given anchor text (i.e. surface form) leading to a corresponding Wikipedia article. At least, when we do not have access to any context of the entity (e.g. when we just see `USA`), the most common meaning of that shortcut is probably the most meaningful match. In [SC12], the authors claim that disambiguation based purely on the commonness of meanings outperforms some of the state of the art methods dealing with the context of entities. However, the most popular or most common meaning is not always the best match and the proper model of an entity context is very important. We can divide the approaches used for named entity disambiguation into two groups: either textual features of a context are compared in order to disambiguate a meaning, or structural relations between entities mentioned in a text are considered.

2.2.1 Textual Disambiguation

Textual representation of an entity context is used in [BPP06]. Links in Wikipedia articles are used as annotations and their surroundings (words within a fixed size window around the annotation) are collected and indexed. They are then compared against the context of a disambiguated entity in new texts. When the context of an entity is not sufficiently big, the taxonomy of Wikipedia categories is taken into account for the disambiguation. For comparison of textual context vectors, the cosine similarity and TF-IDF [RSJ88] weight are used.

Wikify [MC07] and Spotlight [MJGSB11a] use the textual representation of entities described in Wikipedia articles too. Wikify attempts to identify the most likely meaning for a word in a given context based on a measure of contextual overlap between the dictionary definitions of the ambiguous word – here approximated with the corresponding Wikipedia pages, and the context where the ambiguous word occurs (the current paragraph is used as a representation of the context). The approach is inspired by [Les86].

Spotlight represents the context of an entity in a knowledge base by the set of its mentions in individual paragraphs in Wikipedia articles. DBpedia resource occurrences are modeled in a Vector Space Model [SWY75] where each DBpedia resource is a point in a multidimensional space of words. The representation of a DBpedia resource thus forms a meta document containing the aggregation of all paragraphs mentioning that concept in Wikipedia.

The meta document context representation of each candidate entity for an ambiguous surface form is compared to the target paragraph (containing disambiguated entity). The closest candidate in terms of cosine similarity in the vector space model is selected. For weighting individual terms, the TF-ICF weight [MJGSB11a] is introduced. The TF-ICF measure is an adaptation of the TF-IDF [RSJ88] measure. The only difference is that the IDF part is counted among concrete selected candidates and not over the entire knowledge base. Thus, the discriminator terms specific for the concrete candidate selection are weighted higher.

In more recent work [KKR+11], a weakly semi-supervised hierarchical topic model is used for named entity disambiguation. It leverages Wikipedia annotations to appropriately bias the assignment of entity labels to annotated words (and un-annotated words co-occurring with them). In other words the frequency of occurrence of the concrete form of the word in annotations of particular entities in Wikipedia is taken into account, when selecting the correct entity. The Wikipedia category hierarchy is leveraged to capture entity context and co-occurrence patterns in a single unified disambiguation framework.

2.2.2 Structural Disambiguation

In [MW08a], the structure of links to Wikipedia articles corresponding to disambiguated entities is analysed. Each entity is represented by a Wikipedia article. The most similar entities to entities which are

not ambiguous in the texts get higher score. The similarity [MWM08] between two entities represented by Wikipedia articles depends on the number of Wikipedia articles that link to both of them. The score computed this way is then combined with an overall entity commonness for a particular surface form using a C4.5 classifier.

A very similar approach to word sense disambiguation was proposed in [NV05]. WordNet [Mil95] is used as the knowledge base. The disambiguation starts with non-ambiguous words in the text and searches for senses that are connected to these non-ambiguous words. The grammar for this kind of disambiguation is proposed.

A more general approach to structural disambiguation of word senses is introduced in [Mih05]. Distance between candidate labels or senses is counted and a graph is constructed consisting of labels as vertices and distances as weights of edges. The Random Walk adaptation in the form of PageRank algorithm is used to determine scores for individual labels. For each word, its label with the best score is selected. Various representation of distance measures are proposed. For the evaluation, the definition overlap of individual label definitions in a dictionary is used. This sense similarity measure is inspired by the definition of the Lesk algorithm [Les86]. Word senses and definitions are obtained from the WordNet sense inventory [Mil95].

The work presented in [MW08a] was further improved in [KSRC09]. An annotation is scored based on two types of features: one set is local to the occurrence of the surface form of mentioned entity and the other set of features is global to the text fragment. The annotation process is modeled as a search for the mapping that maximizes the sum of the local and global scores of the selected annotations. Experiments over a manually annotated dataset showed that the approach presented in [KSRC09] yields a precision comparable to [MW08a] but outperforms it in terms of recall.

2.2.3 Summary

As LinkedTV focuses on disambiguating named entities in order to retrieve additional content and to obtain additional background knowledge about those named entities, we will favor the approaches using Wikipedia or DBpedia [MC07, MJGSB11a, MW08a, MWM08] since their knowledge base seem to be ideal for this purpose. Wikipedia is one of the biggest freely available knowledge bases on the web. It is also relatively up-to-date, as new concepts (e.g. new products, celebrities, companies) appear relatively early in Wikipedia. Wikipedia is also a general knowledge base which fits into the wide variety of LinkedTV scenarios. URLs of Wikipedia articles can be easily translated to URIs of entities in DBpedia [BLK+09] which provides another valuable source of information about identified entities – in this case in a structured form of RDF documents. Last but not least, Wikipedia is available in the comprehensive extent in all language variations considered within the LinkedTV project.

In LinkedTV, we consider the combination of representative approaches from both groups – namely the approach of DBpedia Spotlight [MJGSB11a] for textual representation of entity context and the structural representation of entity context proposed in [MW08a] together with overall popularity measure [MWM08, SC12]. Our preliminary experiments show that these methods do not overlap and can provide complementary results. The proposal of concrete combination of these method and the evaluation is subject of our future work. We also plan to propose and evaluate a new structure based approach to entity disambiguation.

2.3 NER Web APIs

Recently, serveral web APIs for named entities recognition and disambiguation have been proposed, such as: AlchemyAPI[5], DBpedia Spotlight[6], Evri[7], Extractiv[8], Lupedia[9], OpenCalais[10], Saplo[11], Wikimeta[12], Yahoo! Content Analysis (YCA)[13] and Zemanta[14].

They represent a clear opportunity for the Linked Data community to increase the volume of interconnected data. Although these tools share the same purpose – extracting semantic units from text –

[5] http://www.alchemyapi.com
[6] http://dbpedia.org/spotlight
[7] http://www.evri.com/developer/index.html
[8] http://extractiv.com
[9] http://lupedia.ontotext.com
[10] http://www.opencalais.com
[11] http://saplo.com
[12] http://www.wikimeta.com
[13] http://developer.yahoo.com/search/content/V2/contentAnalysis.html
[14] http://www.zemanta.com

they make use of different algorithms and training data. They generally provide a potential similar output composed of a set of extracted named entities, their type and potentially a URI disambiguating each named entity. The output vary in terms of data model used by the extractors. These services have their own strengths and shortcomings but, to the best of our knowledge, few scientific evaluations have been conducted to understand the conditions under which a tool is the most appropriate one. This section attempts to fill this gap. We have published the results in [RTHB12].

The NE recognition and disambiguation tools vary in terms of response granularity and technology used. As granularity, we define the way how the extraction algorithm works: One Entity per Name (OEN) where the algorithm tokenizes the document in a list of exclusive sentences, recognizing the full stop as a terminator character, and for each sentence, detects named entities; and One Entity per Document (OED) where the algorithm considers the bag of words from the entire document and then detects named entities, removing duplicates for the same output record *(NE, type, URI)*. Therefore, the result set differs from the two approaches.

Table 2 provides an extensive comparison that take into account the technology used: algorithms used to extract NE, supported languages, ontology used to classify the NE, dataset for looking up the real world entities and all the technical issues related to the online computation such as the maximum content request size and the response format. We also report whether a tool provides the position where an NE is found in the text or not. We distinguish four cases: *char offset* considering the text as a sequence of characters, it corresponds to the char index where the NE starts and the length (number of chars) of the NE; *range of chars* considering the text as a sequence of characters, it corresponds to the start index and to the end index where the NE appears; *word offset* the text is tokenized considering any punctuation, it corresponds to the word number after the NE is located (this counting does not take into account the punctuation); *POS offset* the text is tokenized considering any punctuation, it corresponds to the number of part-of-a-speech after the NE is located.

We performed an experimental evaluation to estimate the max content chunk supported by each API, creating a simple application that is able to send to each extractor a text of 1KB initially. In case that the answer was correct (HTTP status 20x), we performed one more test increasing of 1 KB the content chunk. We iterated this operation until we received the answer “text too long”. Table 2 summarizes the factual comparison of the services involved in this study. The * means the value has been estimated experimentally (as the content chunk), + means a list of other sources, generally identifiable as any source available within the Web, finally *N/A* means not available.

	AlchemyAPI	DBpedia Spotlight	Evri	Extractiv	Lupedia	OpenCalais	Saplo	Wikimeta	YCA	Zemanta
Granularity	OED	OEN	OED	OEN	OEN	OED	OED	OEN	OEN	OED
Language support	English French German Italian Portuguese Russian Spanish Swedish	English German (partial) Portuguese (partial) Spanish (partial)	English Italian	English	English French Italian	English French Spanish	English Swedish	English French Spanish	English	English
Restriction on academic use (calls/day)	30000	unlimited	3000	1000	unlimited	50000	1333	unlimited	5000	10000
Sample clients	C/C++ C# Java Perl PHP-5 Python Ruby	Java Javascript PHP	Action Script Java PHP	Java	N/A	Java	Java Javascript PHP Python	Java Perl	Javascript PHP	C# Java Javascript Perl PHP Python Ruby
API interface	CLI JAX-RS SOAP	AJAX CLI JAX-RS SOAP	AJAX JAX-RS	AJAX CLI JAX-RS	CLI JAX-RS	AJAX CLI JAX-RS SOAP	AJAX CLI JAX-RS	CLI JAX-RS	JAX-RS CLI	AJAX CLI JAX-RS
Content chunk	150KB*	452KB*	8KB*	32KB*	20KB*	8KB*	26KB*	80KB*	7769KB*	970KB*
Response format	JSON Microformats XML RDF	HTML+uF(rel-tag) JSON RDF XHTML+RDFa XML	GPB HTML JSON RDF	HTML JSON RDF XML	HTML JSON RDFa XML	JSON Microformats N3 Simple Format	JSON	JSON XML	JSON XML	JSON WNJSON RDF XML
Entity type number	324	320	300*	34	319	95	5	7	13	81
Entity position	N/A	char offset	N/A	word offset	range of chars	char offset	N/A	POS offset	range of chars	N/A
Classification ontologies	Alchemy	DBpedia FreeBase Schema.org	Evri	DBpedia	DBpedia	OpenCalais	Saplo	ESTER (partial)	Yahoo	FreeBase
Deferencable vocabularies	DBpedia Freebase US Census GeoNames UMBEL OpenCyc YAGO MusicBrainz CIA Factbook CrunchBase	DBpedia	Evri	DBpedia	DBpedia LinkedMDB	OpenCalais	N/A	DBpedia Geonames CIAFactbook Wikicompanies others+	Wikipedia	Wikipedia IMDB MusicBrainz Amazon YouTube TechCrunch MusicBrainz Twitter MyBlogLog Facebook others+

Table 2: Factual information about 10 extractors under investigation

2.4 Benchmarking Initiatives and NER Comparison attempts

2.4.1 NER Web APIs Comparison

The creators of the DBpedia Spotlight service have compared their service with a number of other NER extractors (OpenCalais, Zemanta, Ontos Semantic API[15], The Wiki Machine[16], AlchemyAPI and M&W's wikifier [MW08b]) according to a particular annotation task [MJGSB11b]. The experiment consisted in evaluating 35 paragraphs from 10 articles in 8 categories selected from the "The New York Times" and has been performed by 4 human raters. The final goal was to create wiki links. The experiment showed how DBpedia Spotlight overcomes the performance of other services to complete this task. The "golden standard" does not adhere to our requirement because it annotates unit information with just Wikipedia resource and it does not link the annotation to the NE and their type. For this reason, we differentiate from this work by building a proposal for a "golden standard" where we combine NE, type and URI as well as a relevance score of this pattern for the text.

Other attempts of comparisons are stressed in two blog posts. Nathan Rixham[17] and Benjamin Nowack[18] have both reported in their blogs their experiences in developing a prototype using Zemanta and OpenCalais. They observe that Zemanta aims at recommending "tags" for the analyzed content while OpenCalais focuses on the extraction of named entities with their corresponding types. They argue that Zemanta tends to have a higher precision for real things while the performance goes down for less popular topics. When OpenCalais provides a Linked Data identifier or more information about the named entity, it rarely makes a mistake. OpenCalais mints new URIs for all named entities and sometimes provides `owl:sameAs` links with other linked data identifiers. In contrast, Zemanta does not generate new URIs but suggests (multiple) links that represent the best named entity in a particular context. In another blog post, Robert Di Ciuccio[19] reports on a simple benchmarking test of five NER APIs (OpenCalais, Zemanta, AlchemyAPI, Evri, OpenAmplify and Yahoo! Term Extraction) over three video transcripts in the context of ViewChange.org. The author argues that Zemanta was the clear leader of the NLP API field for his tests, observing that OpenCalais was returning highly relevant terms but was lacking disambiguation features and that AlchemyAPI was returning disambiguated results but that the quantity of entities returned was low. Finally, Veeeb provides a simple tool enabling to visualize the raw JSON results of AlchemyAPI, OpenCalais and Evri[20]. Bartosz Malocha developed in EURECOM a similar tool for Zemanta, AlchemyAPI and OpenCalais[21]. We conclude that to the best of our knowledge, there have been very few research efforts that aim to compare systematically and scientifically Linked Data NER services. Our contribution fills this gap. We have developed a framework enabling the human validation of NER web services that is also capable to generate an analysis report under different conditions (see Section 4.2).

2.4.2 Word Sense Disambiguation

Word sense disambiguation (WSD) is a discipline closely related to NER. We analyze the possibility to use Senseval data for benchmarking entity classification systems developed within LinkedTV. Senseval[22] is a series of evaluation exercises for Word Sense Disambiguation. Five Senseval contests were held to date. The first Senseval contest focused on a limited number of generic words across different parts of speech. For nouns, only 15 generic nouns for the English *lexical sample task* such as "accident" or "float" are present [KR00]. For Senseval 2, the general character of the training senses for the lexical sample task is similar to Senseval 1. Senseval 2 and 3 also feature the *all-words task*, where the aim is to disambiguate all words, rather than a sample of selected words. In Senseval 3 approximately 5,000 words of coherent Penn Treebank text are tagged with WordNet 1.7.1 tags. Unfortunately, the selected text contains virtually no named entities. The generic character of words covered applies to all Senseval WSD tasks, including the following Senseval 2007 and SemEval 2010 "Word Sense Disambiguation on a Specific Domain" task. A generic set of words is clearly not suitable for our entity classification problem.

[15] `http://www.ontos.com`
[16] `http://thewikimachine.fbk.eu/`
[17] `http://webr3.org/blog/experiments/linked-data-extractor-prototype-details/`
[18] `http://bnode.org/blog/2010/07/28/linked-data-entity-extraction-with-zemanta-and-opencalais`
[19] `http://blog.viewchange.org/2010/05/entity-extraction-content-api-evaluation/`
[20] `http://www.veeeb.com/examples/flex/nlpapicompare/nlpCompare.html`
[21] `http://entityextraction.appspot.com/`
[22] `http://www.senseval.org`

2.4.3 NER Benchmark Initiatives

The Natural Language Processing (NLP) community has been addressing the NER task for the past few decades, with two major guidelines: establishing standard for various tasks, and metrics to evaluate the performances of algorithms. Scientific evaluation campaigns, starting in 2003 with CoNLL, ACE (2005, 2007), TAC (2009, 2010, 2011, 2012), and ETAPE in 2012 were proposed to involve and compare the performance of various systems in a rigorous and reproducible manner. Various techniques have been proposed along this period to recognize entities mentioned in text and to classify them according to a small set of entity types. We will show how we have used those benchmarks in order to evaluate the NERD platform presented in the section 4.2.

3 State of the Art in Retrieving Additional Content from the Web

The task of retrieval of additional content from the web has generally the following phases:

- determination of types of content to be retrieved,
- identification of sets of possible web resources where appropriate content is available,
- obtaining the content, typically through crawling,
- indexing the content locally, to make the content available for statistical processing,
- search for content relevant to a query entity or video fragment.

In this section, we give the first part of the state-of-the art review of approaches to retrieving additional content from the web, which is focused on the identification of sources of information and programmatic techniques for obtaining content from these data sources. This state-of-the-art review will be complemented with indexing and search phases in subsequent WP2 deliverables. While retrieving additional content from the Web, we consider two main types of content that can be linked to videos:

- Unstructured sources include textual and non-textual data, that do not have a predefined data-model. Textual resources such as ordinary Web pages and Web resources are readable for human, but it is often difficult to cope with them automatically. Similarly, multimedia files (images, videos) can be interpretated by human but are hardly manageable by computers.
- Structured sources of rough data are not very interesting for end users in their original form. They can however serve for enrichment of unstructured content and enhance filtering and recommendation possibilities. They can also support automatic content generation and annotation and include among all Linked Data [BHBL09] resources such as DBpedia [BLK+09] or Freebase [BEP+08].

Different techniques exist for retrieval of such data: specific techniques for data from the Web (section 3.1) and for semantic data (section 3.2) are reviewed. We also give an overview of public APIs that enable to retrieve information from Web resources programmatically (section 3.3). The last section focuses on retrieval by analysis of visual content.

3.1 Web Pages

The retrieval of content is covered in Section 3.1.1. Section 3.1.2 covers technical issues relating to implementation of crawlers. The problem of extracting structured information from retrieved resources is covered in Section 3.1.3. Information extraction from unstructured texts on a Web page is covered in Section 2.

3.1.1 Crawling

General purpose search engines use Web crawlers to maintain their indices [ACGM+01]. A general architecture of such a general purpose crawler is described in [BP98, Cha02]. There is a demand for universal crawlers to crawl as much data as possible and at the same time to keep locally stored versions of crawled information as fresh as possible. Of course, these are conflicting requirements. Therefore a trade off between freshness and completeness has to be found.

3.1.1.1 Queuing Crawlers maintain a frontier queue, containing links to Web pages that need to be processed. In order to utilize resources more effectively a link ordering in a frontier queue was proposed in [CGMP98, HHMN99] to focus on the more *important* pages. The ordering is performed either based on a PageRank [PBMW99] value or based on an indegree of a Web page represented as a node in a link graph. Another crawling strategy was introduced in [Bur97], which selects 64 hosts at a time and crawls them exhaustively, there is no bias related to page quality.

The metrics proposed in [CGMP98] were further maintained and evaluated in [NW01] and it was shown that a simple breadth-first crawling algorithm will tend to fetch the pages with the highest PageRank. In [HHMN00] authors validate that a random walk with acceptance probability proportional to the inverse of frequency that the crawler has encountered a link to a considered Web page yields a sample of the graph that is statistically representative of the original.

Another variation of working with the frontier queue is the best-N-first approach. The crawler picks top N URLs at a time from the frontier (not only one) and fetches them all. Once all N pages are visited, the newly extracted URLs are merge-sorted into the priority queue, and the cycle is repeated. In [PSMM02] it is shown that the best-N-first crawler with N = 256 is performing very good when compared to other options.

Instead of a static strategy used to maintain the frontier queue, crawlers can dynamically adapt to the current conditions on the Web. InfoSpiders [MB00] works with a population of intelligent agents that exist in Web environment. Feedback from the environment consists of a finite energy resource necessary for agent's survival. Each action has an energy cost, which may be proportional to the size of a fetched page or the latency of page download [DPMM01]. Various algorithms for adaptive topical Web crawlers are evaluated in [MPS04]. A more recent approach [AT] uses learning automata [NT89] to determine an optimal order of the crawls.

3.1.1.2 Keeping Index Up-To-Date The dynamics of Web changes was the subject of many studies, among others [CGM99, FMNW04, KL05, Koe02, KLK06]. In [NCO04] authors report that new pages are created at a rate of about 8% per week, only about 62% of the content of these pages is really new because pages are often copied from existing ones. The link structure of the Web is more dynamic, with about 25% new links created per week. Once created, pages tend to change little so that most of the changes observed in the Web are due to additions and deletions rather than modifications. More recent study [ATDE09] reports higher dynamics of the Web. In their collection only 34% of pages displayed no change during the studied interval. On average, documents that displayed some change did so every 123 hours.

The study [NCO04] additionally found that past change was a good predictor of future change, that page length was correlated with change, and that the top-level domain of a page was correlated with change (e.g., edu pages changed more slowly than com pages). This corresponds also with the more recent research [ATDE09]. The study [FMNW04] also confirmed that the degree of change of a page is a better predictor of future change than the frequency of change.

3.1.1.3 Universal Crawler Implementations An example of a scalable distributed Web crawler is UbiCrawler [BCSV04]. It uses consistent hashing to partition URLs according to their host component across crawling machines, leading to graceful performance degradation in the event of the failure of a crawling machine. UbiCrawler was able to download about 10 million pages per day using five crawling machines.

Recently, a single-process Web crawler IRLbot [LLWL08] was presented. This crawler is able to scale to extremely large Web collections without performance degradation. IRLbot features a *seen-URL* data structure that uses only a fixed amount of main memory, and its performance does not degrade as it grows. IRLbot was running over two months and downloaded about 6.4 billion Web pages. In addition, the authors address the issue of crawler traps, and propose ways to ameliorate the impact of such sites on the crawling process.

From open-source crawlers, we recall Heritrix [MKS$^+$04] - the crawler used by the Internet Archive, written in Java. Its design is similar to earlier crawler Mercator [HNN99, NHNH01]. Heritrix is multi-threaded, but not distributed, and as such suitable for conducting moderately sized crawls. Another very popular crawler is the Nutch crawler [KC04] is written in Java as well. It supports distributed operation and has a highly modular architecture, allowing developers to create plug-ins for media-type parsing, data retrieval, querying and clustering.

3.1.1.4 Focused Crawling Rather than crawling pages from the entire Web, we may want to crawl only pages in certain categories. Chakrabarti et al. [CvdBD99] proposed a focused crawler based on a classifier. The idea is to first build a text classifier using labeled example pages from a training set. Then the classifier would guide the crawler by preferentially selecting from the frontier those pages that appear most likely to belong to the categories of interest, according to the classifier's prediction.

However, browsing only pages that belong to the categories of interest may lead to potentially missing many pages that are not directly linked by another pages falling to a "correct" category. In other words, sometimes it may be beneficial to visit pages from a category that is out of interest in order to obtain more links to potentially interesting Web pages. Therefore another type of focused crawlers - Context-Focused Crawlers - were proposed [DCL$^+$00]. They also use naïve Bayesian classifiers as a guide, but in this case the classifiers are trained to estimate the link distance between a crawled page and a set of relevant target pages. It is shown in [DCL$^+$00] that the context-focused crawler outperforms

the standard focused crawler in experiments. An extensive study with hundreds of topics has provided strong evidence that classifiers based on SVM or neural networks can yield significant improvements in the quality of the crawled pages [PS05].

Several variants of focused crawlers were implemented and evaluated in [BPM09]. These include variants of classic, semantic and learning crawlers. Particular emphasis is given to learning crawlers based on the Hidden Markov Model (HMM) [LJM06] capable of learning not only the content of target pages (as classic focused crawlers do) but also paths leading to target pages. Focused crawler using Hidden Markov Models and Conditional Random Fields probabilistic models is proposed in [LMJ04].

3.1.1.5 Topical Crawling For many crawling tasks, labeled examples of pages are not available in sufficient numbers to train a focused crawler before the crawl starts. Instead, we typically have a small set of seed pages and a description of a topic of interest to a user. The topic can consist of one or more example pages or even a short query. Crawlers that start with only such information are often called topical crawlers [Cha03, MPS04]. An example of such topical crawler is MySpiders [PM02]. MySpiders does not maintain any index. It just crawls Web resources at the query time.

The majority of crawling algorithms in the literature are variations of the best-first scheme whereas they differ in the heuristics that they use to score unvisited URLs. A most straight forward approach to computing the score is to count the content similarity between the topic description and the page. The similarity can be measured with the standard cosine similarity, using TF or TF-IDF [RSJ88] term weights.

One alternative to using the entire page or just the anchor text as context, is a weighted window where topic keywords occurrences near the anchor count more toward the link score than those further away [MB00, CDK+99].

3.1.1.6 Deep Web Another important concern by Web resources crawling is the Deep Web [Ber01]. The question, how to access content that is not directly linked from any other Web page by a static link. In other words, how to access the content "hidden" behind Web forms or JavaScript. According to [HPZC07, ZCNN05] the Deep Web has an order of magnitude more data than the currently searchable World Wide Web. A survey covered by Google [MJC+07] indicated that there are in the order of tens of million high-quality HTML forms.

There are two common approaches to offering access to Deep Web content [MKK+08]. The first approach (essentially a data integration solution) is to create vertical search engines for specific domains (e.g. cars, books, or real estate). The integration is achieved by automatically identifying similar form inputs and selecting the best mediated form [HC06, WYDM04, WWLM04].

The second approach is surfacing, which pre-computes the most relevant form submissions for all interesting HTML forms [BFF04, Nto05]. The URLs resulting from these submissions are generated offline and indexed like any other HTML page.

3.1.2 Robots.txt and Load Balancing

The Robots Exclusion Protocol[23] published inside robots.txt files is a valuable resource of information for crawlers (an example of such file is displayed in Figure 4). The term robot is used as a synonym for the crawler here. Crawlers are often marked also as bots such as Googlebot by Google crawler, Bingbot for Bing search engine and Yahoo! Slurp for Yahoo! search. The information in robots.txt is not limited only to crawlers of web pages. Also crawlers consuming other types of data (such as images or other types of multimedia) should respect the rules provided in robots.txt.

In order not to crawl sensitive information and not to overload crawled servers it is important to respect *disallowed* pages and information about crawling delays between requests to one host.

Especially structured data sources introduce very often restrictive crawl delays in order to avoid overload of the server by extensive crawlers. For example DBpedia prescribes 10 seconds crawl delay[24] which slows the crawler significantly down. However, in this case aggregated dump files are provided, so there is no need for crawling the whole Web.

Additionally, Web authors can indicate if a page may or may not be indexed, cached, or mined by a crawler using a special HTML meta-tag. Crawlers need to fetch a page in order to parse this tag, therefore this approach is not widely used.

[23] More details on the robot exclusion protocols can be found at `http://www.robotstxt.org/orig.html`.
[24] http://dbpedia.org/robots.txt

```
# Hash symbols indroduces a comment

# Restriction for all robots
User-agent: *
Disallow: /private/

# Restriction only for GoogleBot
User-agent: GoogleBot
Disallow: /not/for/google/

# Page specificaly set as allowed for crawling robots
Allow: /google/info.html

# Specific location of a host sitemap resource
Sitemap: http://example.com/sitemap.xml

# Prescribed crawl delay 10 seconds between requests
Crawl-Delay: 10
```

Figure 4: Robots.txt example

3.1.3 Wrappers

Wrappers as tools for extraction of structured data from Web resources (among all Web pages) can be divided into three groups according to techniques of wrapper generation [Liu07a]:

1. Manually created wrappers (includes wrapper programming languages and visual tools helping users to construct wrappers)
2. Automatic wrapper induction
3. Automatic data extraction

3.1.3.1 Manually Created Wrappers The representatives of this group include academic projects such as Lixto [BFG01], XWRAP [LPH00] and Wargo [RPA+02, PRA+02] as well as commercial systems Design Studio (originally RoboMaker) by Kapow technologies[25] and WebQL[26] by QL2 Software. They focus on methods of strongly supervised "semi-automatic" wrapper generation, providing a wrapper designer with visual and interactive support for declaring extraction and formatting patterns.

3.1.3.2 Automatic Wrapper Induction This group focuses on automatic wrapper induction from annotated examples and includes WIEN [Kus97a], Stalker [MMK99], DEByE [LRNdS02], WL^2 [ZL05] or IDE [CHJ02]. Most automatically induced wrappers learn patterns from a set of previously labeled examples. IDE [CHJ02] reduces the effort needed for manual annotation by learning on the fly and requiring manual annotation only by Web pages that can not be processed with previously learned rules. The aim is to minimize the unnecessary annotation effort.

Most existing wrapper induction systems build wrappers based on similar pages assuming that they are generated from the same template. In [ZNW+05] the two-dimensional Conditional Random Fields model is used to incorporate two-dimensional neighborhood dependencies of objects at the Web page. The system learns from labeled pages from multiple sites in a specific domain. The resulting wrapper can be used to extract data from other sites. This avoids the labor intensive work of building a wrapper for each site.

Whereas Web wrappers dominantly focus on either the flat HTML[27] code or the DOM[28] tree representation of Web pages, recent approaches aim at extracting data from the CSS[29] box model and, hence, the visual representation of Web pages [GBH+07]. The visual representation of a Web page

[25] http://kapowsoftware.com/
[26] http://www.ql2.com/products-services/ql2-webql/
[27] HyperText Markup Language
[28] Document Object Model
[29] Cascading Style Sheets

enables the wrapper to overcome to some extent changes in HTML code that usually distract HTML or DOM based wrappers.

3.1.3.3 Automatic Data Extraction Finally automatic data extraction approaches try to extract data without any assistance. They aim at extracting data records from data regions identified on a Web page – for example in list pages [EJN99, BLPP01, CL01].

The MDR algorithm discussed in [LGZ03] uses string edit distance in pattern finding. An algorithm based on the visual information was proposed in [ZMW+05] for extracting search engine results.

DEPTA [ZL06] uses the visual layout of information in the page and tree edit-distance techniques to detect lists of records in a page and to extract the structured data records that form it. DEPTA requires as an input one single page containing a list of structured data records.

In [APR+10] also only one page containing a list of data records as input is needed. The method begins by finding the data region containing the dominant list. Then, it performs a clustering process to limit the number of candidate record divisions in the data region and chooses the one having higher autosimilarity according to edit-distance-based similarity techniques. Finally, a multiple string alignment algorithm is used to extract the attribute values of each data record.

In contrast to approach presented by [APR+10], the RoadRunner [CMMM01] system needs as input multiple pages generated based on the same template. The pages can be either detail pages or list pages. The work of RoadRunner was continued and improved in [AGM03].

3.1.4 Indexing and Retrieval

There are many ready made open source projects focused on indexing and retrieval textual documents implementing state of the art information retrieval techniques (e.g. Lucene [wwwa], Solr [wwwb], Sphinx [wwwd], MnogoSearch [wwwc]). In the research community, probably the most attention gets Lucene or Lucene based index Solr. These indexes are both powerful and flexible in terms of extension. Even advanced indexes to search structured data like Siren [Del09] were developed based on Lucene or Solr index [HUHD07, HHUD07b, BKvH02, CDD+04].

In [KPT+04] authors argue that named entities mentioned in the documents constitute important part of their semantics. KIM platform [PKK+03] introduces rather a structured approach to information retrieval and discovery of related information. Named entities identified in texts are used to provide more precise search results. Thanks to the mapping of discovered entities to a backing ontology, the system is able to respond even to structured queries, not only keyword based queries.

The idea of exploiting semantic annotations for better retrieval was proposed already in earlier works [KKPS02, HSC02, Pas04]. In [CHSS08] probabilistic modelling framework is proposed that combines both human-defined concepts and data-driven topics is used to model the content of documents. Wikipedia‘s concept relatedness information is combined with a domain ontology to produce semantic content classifiers for content filtering in [MSA+10].

3.1.5 Summary

The crawler will be an important part of the web linking process for web resources that can not be directly queried (using an API or some kind of inducted wrapper). LinkedTV scenarios include curated or white list web site. Therefore, the use of focused crawler techniques rather than general crawling approaches is considered. However, the guidelines for general crawling apply also to focused crawlers (i.e. predicting web page changes [NCO04] for re-crawling or respecting the Robots Exclusion Protocol described in Section 3.1.2). For crawling, we consider the use of the state of the art tools such as Heritrix [MKS+04] or the Nutch crawler [KC04] and their customization for LinkedTV specific needs.

A great source of inspiration for related content linking comes from the KIM platform [PKK+03] where structured information about named entities in texts is used for their advanced filtering. From a technical point of view, the popular Lucene [wwwa] based index Solr [wwwb] can be used and extended in order to index this kind of information for faster retrieval.

We currently do not consider direct usage of wrappers. However, wrapper-based techniques might be used for offline processing of crawled pages and extraction of structured content from them [BFG01, ZNW+05, EJN99, BLPP01, CL01, ZL06]. Important tasks include identification of important regions in a web page (e.g. area of the main text of a news articles without advertisements, comments, etc.) and extraction of particular parts of web pages (e.g. multimedia or images).

3.2 Semantic Web

In this section, we enumerate different types of semantic web resources and the ways to access them. Then, we cover related work in consumption of Semantic Web resources. Finally, we provide an insight in semantic sitemaps that help to explore new sources of information by locating structured sources of data on individual domains.

3.2.1 Types of resources

There are various sources of data available on the Web. In this section, we briefly recall ways of publishing data on the Web.

Annotated Web Pages. Structured information can be embedded directly in any ordinary Web page in the form of annotations. Several format of data presentation have emerged including Microformats (marking data with special classes of html tags – e.g. vCard, hCalendar), eRDF, RDFa (RDF serializations embedded in HTML markup) or Microdata (HTML5 standard for presentation of semantic information in the context of a Web page).

The structured content can be extracted and transformed to a common output format using parsers such as Any23 [any12] (for most of currently used annotation formats, with a modular architecture enabling addition of own custom data extractors written in Java) or Java-RDFa [jav12] (an RDFa extractor written in Java).

RDF Resources and Dumps. A more condensed way of publishing structured data is to provide an RDF file in some of the RDF serialization formats (e.g. RDF/XML[30], N3[31], Turtle[32], N-Triples[33]). Consumption of such RDF files poses less overhead for the crawler, as it can start to consume directly the data and does not have to cope with additional HTML syntax, which is often not valid and difficult to deal with correctly.

Many big Linked Data [BL06] hubs such as DBpedia [BLK+09] or Freebase [BEP+08] provide data dumps in order to save resources needed for crawling of individual data files or Web pages. A massive crawling of individual Web pages may result in the target server overload and the denial of service (see Section 3.1.2).

SPARQL Endpoints. A SPARQL endpoint is a conformant SPARQL protocol service. A SPARQL endpoint enables users to query a knowledge base via the SPARQL language [PS06]. Results are typically returned in one or more machine-processable formats. A SPARQL endpoints are usually accessible via HTTP requests as Web services. Queries that can be performed against a SPARQL endpoint are often limited because of performance reasons. Therefore, crawling all data in a knowledge base via a SPARQL endpoint is often less effective than their consumption in the form of data dumps (if available). However, SPARQL endpoints are very useful in order to obtain a certain type of data or a subset of records.

3.2.2 Crawling.

On the Web, a crawler is an essential part of a search engine [BP98]. Similar situation is on the Semantic Web. Probably one of the first adopters of crawling technologies were authors of Semantic Web search engines. When Watson [SDB+07] was developed, one of the biggest crawling problems was, how to actually discover semantic data resources. The authors proposed several heuristics including exploring well-known ontology repositories and querying Google with special type of queries. The crawling of the discovered content relies on Heritrix crawler[34].

A similar issue was addressed by Swoogle [DFJ+04], where three specific crawlers were developed: *Google Crawler* (for querying Google and crawling search results), *Focused Crawler* for crawling documents within a given Web site and *Swoogle Crawler*, which follows URIs of resources identified in discovered Semantic Web documents. With emergence of Linked Data [BL06] still more and more resources are interlinked and the location of new data sources is not so difficult.

[30] The RDF/XML specification can be found on `http://www.w3.org/TR/REC-rdf-syntax/`

[31] The N3 specification can be found on `http://www.w3.org/TeamSubmission/n3/`

[32] The TurtleL specification can be found on `http://www.w3.org/TR/2011/WD-turtle-20110809/`

[33] The N-Triples specification can be found on `http://www.w3.org/2001/sw/RDFCore/ntriples/`

[34] `http://crawler.archive.org`

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
xmlns:sc="http://.../sitemapextension/scschema.xsd"
>
   <url>
      <loc>http://www.example.org/</loc>
      <lastmod>2012-01-01</lastmod>
      <changefreq>weekly</changefreq>
      <priority>0.9</priority>
   </url>
  <sc:dataset>
    <sc:datasetLabel>
        Example Corp. Product Catalog
      </sc:datasetLabel>
    <sc:datasetURI>
        http://example.com/rdf#catalog
    </sc:datasetURI>
    <sc:linkedDataPrefix slicing="subject-object">
        http://example.com/products/
    </sc:linkedDataPrefix>
    <sc:sampleURI>
        http://example.com/products/X42
  </sc:sampleURI>
    <sc:sparqlEndpointLocation slicing="subject-object">
      http://example.com/sparql
    </sc:sparqlEndpointLocation>
    <sc:dataDumpLocation>
      http://example.com/data/catalogdump.rdf.gz
    </sc:dataDumpLocation>
    <changefreq>weekly</changefreq>
  </sc:dataset>
</urlset>
```

Figure 5: Sitemap.xml example with Sitemap Semantic extension data

A pipelined crawling architecture was proposed for MultiCrawler [HUD06] employed in SWSE semantic search engine [HHUD07a, HHD+07]. MultiCrawler deals also with performance scaling. It was achieved by distributing processed pages to individual computers based on a hashing function. However, the authors do not deal with a fail over scenario, where some computers in the cluster might break down.

A multithreaded crawler was used to obtain data for Falcons search engine [CQQ09]. A more sophisticated crawling infrastructure is employed in Sindice project [ODC+08]. It proposes a processing pipeline similar to SWSE, but uses a parallel Hadoop[35] architecture. The authors propose also a semantic sitemap format, which is an variation of sitemap format for ordinary crawlers of unstructured content adapted for structured data needs.

LDSpider [IHU+10] is a Java project that enables performing custom crawling tasks. The spider performs concurrent crawling by starting multiple threads. However, all the threads still use shared CPU, memory and storage.

3.2.3 Semantic Sitemaps

Another valuable source of crawling information are sitemap XML files (an example sitemap is shown in Figure 5). There are three types of data sources that can be discovered in a semantic sitemap:

- URL of individual resources
- Sitemap index files
- Semantic data dump locations

Individual resources contain usually information about one or a limited count of entities. Often these are ordinary Web pages annotated with some semantic data or locations of small RDF files describing a concrete entity. Sitemap index files provide information where to find more sitemaps on the same host, because sometimes sitemaps are so large that they have to be split into multiple files. Data dump locations are gold-mines for crawlers. They contain sets of RDF triples, which were already collected from the host and aggregated by its owner.

[35] Apache Hadoop is an open-source software for reliable, scalable, distributed computing. More information can be found on the website of the project `http://hadoop.apache.org/`

3.2.4 Summary

Crawling semantic web resources has some specificities. The ways of obtaining information from Semantic Web resources are quite diverse (see Section 3.2.1). Therefore the crawler should support a combined way of data consumption (i.e. combination of querying and crawling), which current semantic web crawlers [HUD06, ODC+08, IHU+10] do not fully support. Additionally, we consider the use of a local cache to store results of crawling, which would be used to support faster querying, since the use direct queries on live Semantic Web resources is limited due to performance issues.

3.3 Retrieval from Public APIs

Public APIs (Application Programming Interfaces) offer interfaces that we can use to communicate with content providers, aggregators or searching websites in our case. Indeed, they make it easier for the developer to retrieve content, i.e. they enable a direct search for specific content through HTTP(S) requests, while avoiding a tedious crawling. A main drawback of using public APIs is that they are black-box services that we do not control. They can be deprecated or closed down at any time by the company owning them, leaving the developer without resource. For instance, when bought by Facebook in 2012, Face.com[36] disabled the access to its service. Also, we should point out that some features may need to be redesigned in order to better match the requirements of the project.

3.3.1 Public Search Services

All three major search engines offer also an automatic access to search results via their Web based APIs. Such an access is very useful for locating relevant information all over the Web.

Google. The Google Custom Search API[37] lets developers to create Web sites and programs to retrieve and display search results from Google Custom Search programmatically. Google provides a RESTful API for requests to get either Web search or image search results in JSON or Atom format. Images can be filtered according to their color, size, type (i.e. clipart, face, lineart, news, and photo) and file type. Web search results can be limited to a certain country or language. The results can be further filtered based on licensing. The usage is free up to 100 requests per day.

Yahoo!. Yahoo! Search BOSS (Build your Own Search Service)[38] provides a RESTful API access to Yahoo!'s Web, news, and image search technology. Yahoo! as well as Google and Bing offers an access to spelling corrections and suggestions. Additionally the BOSS service offers access to Yahoo! specific structured content in Web results (where available) and the possibility to blend and re-rank search results. XML and JSON formats are supported. The API returns results for following source types:

- Web – Yahoo! Web search index results with basic url, title, and abstract data.
- Limited Web – Limited Web results. Index refresh rate 3days.
- Image search – Image Search includes images from the Yahoo! Image Search index and Flickr.
- News search – News Search includes late breaking news articles from the past 30 days.
- Blogs – Blogs search (in Beta version).
- Advertising – If publisher has qualified for Yahoo! Search Advertising.

The pricing starts at 0.1 USD per 1 000 queries.

[36] http://www.face.com
[37] https://developers.google.com/custom-search/v1/overview
[38] http://developer.yahoo.com/search/

Bing. The Bing Search API[39] enables developers to embed and customize search results in applications or Web sites using XML or JSON. The API offers multiple source types (or types of search results). It is possible request a single source type or multiple source types with each query (e.g. Web, images, news, and video results for a single search query). The Bing Search API returns results for the following source types:

- Web – Web search results.
- Images – Image search results.
- News – News search results.
- Videos – Video search results.
- Related Search – Related search suggestions based on the query entered.
- Spelling Suggestions – Spelling suggestions based on the query entered.

The usage is free up to 5 000 requests per month.

3.3.2 Public Search Services for Semantic Web

Apart from ordinary search engines there is also a possibility to locate structured data sources on the Web.

Sindice. Sindice.com[40] is a semantic search engine that offers also the possibility to localize and retrieve structured documents from the Web [ODC+08]. The Sindice API provides the programmatic access to its search capabilities. Supported response formats are JSON, ATOM and RDF/XML. Sindice provides various ways of querying the dataset:

- Fulltext Search – Simple keyword based queries similar to queries posted to ordinary search engines.
- N-Triples Search – Queries passed in a special Sindice Query Language[41].
- SPARQL endpoint – Recently Sindice has made accessible a SPARQL endpoint[42].

Additionally, Sindice Live API allows developers to retrieve triples from Web documents using document uri or content. Sindice is able to perform reasoning on the fly.

SameAs. SameAs.org[43] is a Web service that helps to find co-references between different data sets. If a URI is provided, it will give back URIs that may be co-referent. According to its Web page the service currently covers over 125 million URIs. Locating co-refereces helps by discovery of additional information about same entity or concept in different datasets.

3.3.3 Retrieval of media content

Media content we deal with includes images and videos files. First, there exists some APIs for content retrieval from specific image or video sharing platforms. A general use is to browse, access, publish and modify data. In particular, they enable users to make a search for data on the platform following several criteria such as keyword/tag, location, category or user. It is also possible to retrieve the most popular content or browse content by exploring relationships between objects (as those platforms act as networks). Other uses, out of the scope of this document, include getting information on users, dealing with the user's albums or commenting on images/videos. Depending on the action made and on the API used, an authentication phase is needed or not.

[39] http://www.bing.com/developers/
[40] http://www.sindice.com
[41] http://sindice.com/developers/queryLanguage
[42] http://sparql.sindice.com/
[43] http://sameas.org/

Picasa. Picasa Web Albums Data API[44] is the API that integrates with Picasa Web Albums from Google. It allows users to deal with photo albums, comment other photos or make searches on content. It uses the RESTful Google Data Protocol, which enables applications to access and update the data stored in Google products. Results are either Atom (by default) or JSON feeds.

Instagram. Instagram is a mobile application for photo editing and sharing. Its API[45] offers a real-time update of results using part of the Pubsubhubub protocol, that notifies the created system of new content posted. The results are published in JSON with a limit of 5000 requests per hour per user.

Flickr. Flickr from Yahoo provides a very complete API[46]. Flickr is an online photo management and sharing application. This API supports diverse request formats (REST, XML-RPC, SOAP) and diverse response formats (REST, XML-RPC, SOAP, JSON, PHP). The query limit is of 3600 per hour per key.

Youtube. Youtube's Data API[47] is the API the enables a user to perform operations similar to those available on the Youtube website. Similarly as Picasa Web Albums Data API, it uses the Google Data Protocol for retrieval of feeds about media and users. It offers a courtesy limit of 5,000 requests/day.

Vimeo. Vimeo APIs[48] are of two types: the simple API, read-only and limited to public data, and the Advanced API for tasks that need authentication. Search of videos can be made with the Advanced API only (the simple API only allows browsing of videos by channel, user or group). Results are returned using either JSON, XML or PHP format. Limits of usage for search apply but are not disclosed: they are presumed "adequate for most apps" according to the API guidelines.

Dailymotion. Dailymotion's Graph API[49] presents a view of Dailymotion as a graph of connected objects (users, videos, playlists, etc). Requests (queries on objects or relationships) can be made on the graph; response are JSON objects. For more control over latency and caching, it is possible to use the Advanced API[50].

Wikimedia. MediaWiki API[51] enables the user to access data contained in the MediaWiki databases, among which we find Wikipedia (free collaborative encyclopedia) and Wikimedia Commons (repository of freely usable media files). The API processes with RESTful calls and supports diverse response formats (XML, JSON, PHP, YAML and others).

Europeana. Europeana is an online portal that gives access to digitalized content (books, painings, films, etc) from diverse European heritage institutions, by storing contextual information about the items, including a picture and a link to the actual content in the original institution website. Europeana API[52] allows a search over those objects based on the Open Search standard. Europeana API services are only available to Europeana network partners.

While the previous APIs comes from specific media sharing platforms with social activity, social networks enable the user to share different kinds of media. Depending on the plateform, there are two different ways to share media content: either the user posts a comment containing a link to an external content, or (s)he directly uploads the content to her/his account.

Facebook. Facebook Graph API[53] depicts Facebook data as a graph of connected objects (user, video, image, album, comment, post, etc) with a unique id. The API allows to search the graph, giving responses as JSON objects. It supports real-time updates in order for the application to be aware of changes.

[44] https://developers.google.com/picasa-web/
[45] http://instagram.com/developer/
[46] http://www.flickr.com/services/api/
[47] https://developers.google.com/youtube/
[48] http://developer.vimeo.com/apis
[49] http://www.dailymotion.com/doc/api/graph-api.html
[50] www.dailymotion.com/doc/api/advanced-api.html
[51] http://www.mediawiki.org/wiki/API
[52] http://pro.europeana.eu/reuse/api
[53] https://developers.facebook.com/docs/reference/api/

Google+. Google+ API[54] is the API on top of Google+ social network. It provides read-only access to data, structured in three different resource type: people, activities (notes posted, that can include images, videos, links) and comments. On a technical point of view, it follows a RESTful design and uses JSON format to represent resources. The courtesy usage limit is of 10,000 requests/day.

Twitter. The micro-blogging service provides a REST API[55] composed of two parts: the actual so-called REST API and the Search API (which is also RESTful) that are planned to get unified in the future. The Search API enables search of recent tweets (no later than 9 days) and needs no authentication. It returns results using JSON or Atom, while the REST API supports XML, JSON, RSS and Atom. The Twitter REST API methods allow developers to access core Twitter data, hence enabling exploration of the network as such (exploring user information, relationships between users, tweets, etc).

It also includes the Streaming API[56] which requires an architecture different from a REST API: it needs a persistent HTTP connection open to stream the data real-time. It is more suitable when building a mining application with intensive data needs. The Streaming API allows for large quantities of keywords to be specified and tracked, retrieving geo-tagged tweets from a certain region, or have the public statuses of a user set returned. The messages streamed are JSON objects. Media content is not directly embedded in the twitter messages, but these twits may contain links to media content, which we can retrieve by following the links.

3.4 Retrieval through analysis of visual content

The scope of this project is to retrieve multimedia content related to a given video. This process includes a ranking phase in order to give priority to the content considered as *most relevant*. We are focusing on visual content retrieval, namely images and videos (audio content is not considered for retrieval). Videos are a sequence of images on a linear time basis. Hence, we will review retrieval of content from image data in this section.

Clearly, dealing with visual content is more complex than dealing with textual content: it is not straightforward to capture information in the form of semantic concepts from an image. While human tend to classify and search for images given high-level features, computer vision techniques mostly rely on low-level features that have limited descriptive power. This is called the *semantic gap* [BYRN11]. While words are basic units that are readily interpretable, semantic unit in visual content is a parameter that needs to be chosen when designing the search algorithm. It will heavily affect the results returned. The structure of data is also different: while text can be seen as linear data, images and videos have higher dimensions, because of spatial (the image frame) and temporal (in videos) dimensions. Last, image understanding can be subject to personal interpretation.

A typical content-based retrieval system stores images in a database after extraction of low-level features. A similarity measure with the query is then defined (proper to the system) that enables the search with a given algorithm. Then, a user feedback step can be added to refine the results and improve the relevance of the results.

3.4.1 Low-level features analysis

Low-level features are used to describe visual content; their extraction is the first step to retrieval by content. Descriptors can mainly be divided into two categories: local and global ones. The former refers to a group of descriptors that computes local features on regions of the image, while the latter gathers descriptors that use global features, computed on the whole image. In particular, some work of the the MPEG (Moving Picture Experts Group) led to release the MPEG-7 standards features for multimedia content description.

Global descriptors include color and texture descriptors. The color distribution of an image is captured through color histograms. MPEG-7 proposes different descriptors: the dominant color descriptor (DCD), the color layout descriptor(CLD), the color structure descriptor (CSD), the scalable color descriptor (SCD) and the group of frame (GoF) or group of picture (GoP) descriptor. Texture is a measure of the patterns of intensity; widely used descriptors are: the homogeneous texture descriptor (HTD), the edge histogram descriptor (EHD), the texture browsing descriptor (TBD). Shape is another feature of interest. MPEG-7 descriptors are the region-based shape descriptor (R-SD), the contour-based shape descriptor

[54] https://developers.google.com/+/api/
[55] https://dev.twitter.com/docs/api
[56] https://dev.twitter.com/docs/streaming-apis

(C-SD) and the 3D shape descriptor (3D-SD). We can also cite the Edge Orientation Histograms (EOH) and the Histogram of Oriented Gradients (HOG). For more details, see [THS11].

Local descriptors describe image content localized on a particular regions of the image. In order to compute local features, the image first have to be divided into smaller areas, i.e. regions of interest have to be determined. Detectors of edges or corners are used for such tasks, for instance the Harris corner detector. Then, global descriptors can be used to describe those regions; otherwise, specific local descriptors exist. SIFT (Scale-invariant feature transform) and SURF (Speeded Up Robust Feature) are particularly widely-used for finding consistent features over a change of scale or viewpoint. A more comprehensive review can be found at [LA08]. Most of this work has been reviewed by WP1 (see D1.1 for more details on descriptors).

3.4.2 Image retrieval systems

When retrieving related content, two sorts of behavior are possible:

- The user can be looking for the same content, i.e. (s)he wants to identify multiple copies of the same content : it is the duplicate or near-duplicate problem. This would be useful to identify different news programs using the same image for example, nevertheless it is not in the scope of this project.
- Instead, (s)he could be looking for content that is *similar* but still different. The idea is to analyze images in order to extract features or concepts that we want to find in additional content.

Hence, similarity is a key concept that has to be defined; it is at the core of the retrieval system. Similarity can be based on low-level features (images of scenes that have a similar color distribution, thus are visually similar at first sight), or on semantics (images that contains semantically close concepts).

The main components of a retrieval system are: the model to represent the images from the low-level features extracted, the type of query, and the similarity measure used to rank the retrieved content. A relevance feedback phase can be added to progressively refine the results after user's marking images as relevant or not to her/his query. Their choice defines the type of content retrieved.

The model. Typically, after extraction of low-level features, the image is described using an image signature, typically a vector or a distribution. Assessing the similarity is based on this description.

Early works were using multi-dimensional feature vectors for image representations, and were comparing them using diverse measures, among which: the Manhattan distance, the Euclidean distance, the Mahanalobis distance [Mah36] and the Earth-mover's distance [RTG98]. IBM's QBIC system uses such a distance calculation between feature vectors [FSN+95].

Following research on textual data, [SMMP00] indexes images as a vector of low-level features using an inverted file. Weigthing of terms is applied as well for vector matching, based on the frequency of occurrence of features in the entire collection.

Later, the Bag-of-Words approach has been inspired from research on text retrieval. This technique treats images as documents; a codebook is generated from the features and images are indexing according to this codebook. A main drawback of this approach is that the representation does not take into account spatial relationships of the image features. Sivic and Zisserman describe objects in videos as "visual words" using SIFT features and k-means clustering, that build a vocabulary. Images are represented and retrieved using the "bag-of-words" approach, i.e. a weighted vector of visual words frequency [SZ03].

In order to bridge the semantic gap, the common approach is now to automatically annotate images with semantic label of concepts that can be detected, thanks to trained models using machine learning tools. Indeed, higher level semantics can be trained from collected samples and then used to annotate new images. For a comprehensive review on such techniques, refer to [ZIL12]. Another trend is to incorporate metadata coming from associated text, for example text surrounding an image in a Web page. Such methods are called "hybrid methods". The Cortina system ([QMTM04])is a large-scale retrieval system that combines text-based (high-level semantics search) and content-based (based on low-level descriptors) searches. First is an offline phase where crawling is performed; indexing by keywords in an inverted file index (based on associated textual content) is complemented by an indexing in a relational database of clustered visual features. The search is performed by keyword, and is refined based on visual attribute after relevance feedback from the user.

The query. The user's query that can be of different types:

- Query-by-example is the most popular one. The query is made by providing an example image of the content the user wants to look for. The image is then represented on the same model as the indexed images (after extraction of features), and then matched to the database for similarity.
- The query can also be expressed in terms of the feature representations itself (color or texture queries for instance).
- The user can draw a sketch of the content (s)he is looking for.

The query is then transformed into the representation of the images so a match with the images from the database can be performed: similarities are calculated and used for ranking of the content.

Another possible retrieval system is one where the user is browsing through an ordered collection of images. The images are presented in different clusters that can be refined by the user's search, hence enabling him to navigate through customized categories thanks to visual attributes. Google Image Swirl [JRW+12] creates clusters based on semantic and visual similarities and display them to the user as a tree, thus enabling hierarchical browsing.

4 Entity Recognition and Disambiguation – Requirements and Specification

The primary sources for performing Named Entity Recognition and Disambiguation are the subtitles of the seed videos being watched on the LinkedTV system. Alternatively, another textual source can also be the ASR transcripts generated by WP1 and stored in the eXmaralda format. By nature, those transcripts will be more noisy, often grammatically incorrect depending on the performance of the ASR engine. However, as we will see in the section 4.3, the performance of NER on ASR transcripts are similar than on perfect subtitles using our proposed named entity framework.

4.1 Functional Requirements

As we have see in the Figure 2, WP2 aims at providing links related to the seed video content that will be either directly consumed by the presentation engine (WP3) or being post-processed by the LinkedTV personalization module (WP4). The personalization layer requires that the video shots are described by a number of features that correspond to criteria for which the user (in part subconsciously) applies to assess the degree of interestingness of the particular shot. The input required for WP4 as defined in D4.2 is in the form of crisp or fuzzy description of entities in the shot:

- crisp classification: the entity is categorized with at most one type which is mapped to an ontology concept (NERD ontology) and disambiguated with a LOD resource (in most cases, a DBpedia URI).
- fuzzy classification: the entity is categorized under several types with different scores.

In this section, we describe the Named Entity Recognition and Disambiguation (NERD) framework, our proposal for unifying the output results of the various NER web APIs reviewed in the section 2.3.

4.2 NERD: a Platform for Named Entity Recognition and Disambiguation

NERD is a web framework plugged on top of various NER extractors. Its architecture follows the REST principles [FT02] and includes an HTML front-end for humans and an API for computers to exchange content in JSON. Both interfaces are powered by the NERD REST engine.

4.2.1 NERD Data Model

We propose the following data model that encapsulates the common properties for representing NERD extraction results. It is composed of a list of entities for which a label, a type and a URI is provided, together with the mapped type in the NERD taxonomy, the position of the named entity, the confidence and relevance scores as they are provided by the NER tools. The example below shows this data model (for the sake of brevity, we use the JSON syntax):

```
"entities": [{
    "entity": "Kalifornien",
    "type": "StateOrCounty",
    "nerdType": "http://nerd.eurecom.fr/ontology#Location",
    "uri": "http://de.dbpedia.org/resource/Kalifornien",
    "startChar": 346,
    "endChar": 357,
    "confidence": 0.288741,
    "source": "alchemyapi",
    "startNPT": 79622.9,
    "endNPT": 79627.3
    }]
```

which indicates that “Kalifornien” is a named entity of type `StateOrCounty` for the extractor `AlchemyAPI`, which has been mapped to the type `nerd:Location` and disambiguated with the German DBpedia URI `http://de.dbpedia.org/resource/Kalifornien`. It also indicates that the source of this extraction is `AlchemyAPI` with a confidence score of 0.288741, and that this named entity has been spotted in the transcript of a video in the time range [79622.9, 79627.3] in seconds.

4.2.2 NERD REST API

The REST engine runs on Jersey[57] and Grizzly[58] technologies. Their extensible frameworks enable to develop several components and NERD is composed of 7 modules namely authentication, scraping, extraction, ontology mapping, store, statistics and web. The authentication takes as input a FOAF profile of a user and links the evaluations with the user who performs them (we are developing an OpenID implementation and it will replace soon the simple authentication system working right now). The scraping module takes as input the URI of an article and extracts all its raw text. Extraction is the module designed to invoke the external service APIs and collect the results. Each service provides its own taxonomy of named entity types it can recognize. We therefore designed the NERD ontology which provides a set of mappings between these various classifications. The ontology mapping is the module in charge to map the classification type retrieved to our ontology. The store module saves all evaluations according to the schema model we defined in the NERD database. The statistic module enables to extract data patterns from the user interactions stored in the database and to compute statistical scores such as the Fleiss Kappa score and the precision measure. Finally, the web module manages the client requests, the web cache and generates HTML pages.

Plugged on the top of this engine, there is an API interface[59]. It is developed following the REST principles and it has been implemented to enable programmatic access to the NERD framework. It follows the following URI scheme (the base URI is `http://nerd.eurecom.fr/api`):

`/document` : GET, POST, PUT methods enable to fetch, submit or modify a document parsed by the NERD framework;

`/user` : GET, POST methods enable to insert a new user to the NERD framework and to fetch account details;

`/annotation/{extractor}` : POST method drives the annotation of a document. The parametric URI allows to pilot the extractors supported by NERD;

`/extraction` : GET method allows to fetch the output described in section 4.2.1;

`/evaluation` : GET method allows to retrieve a statistic interpretation of the extractor behaviors.

4.2.3 NERD Ontology

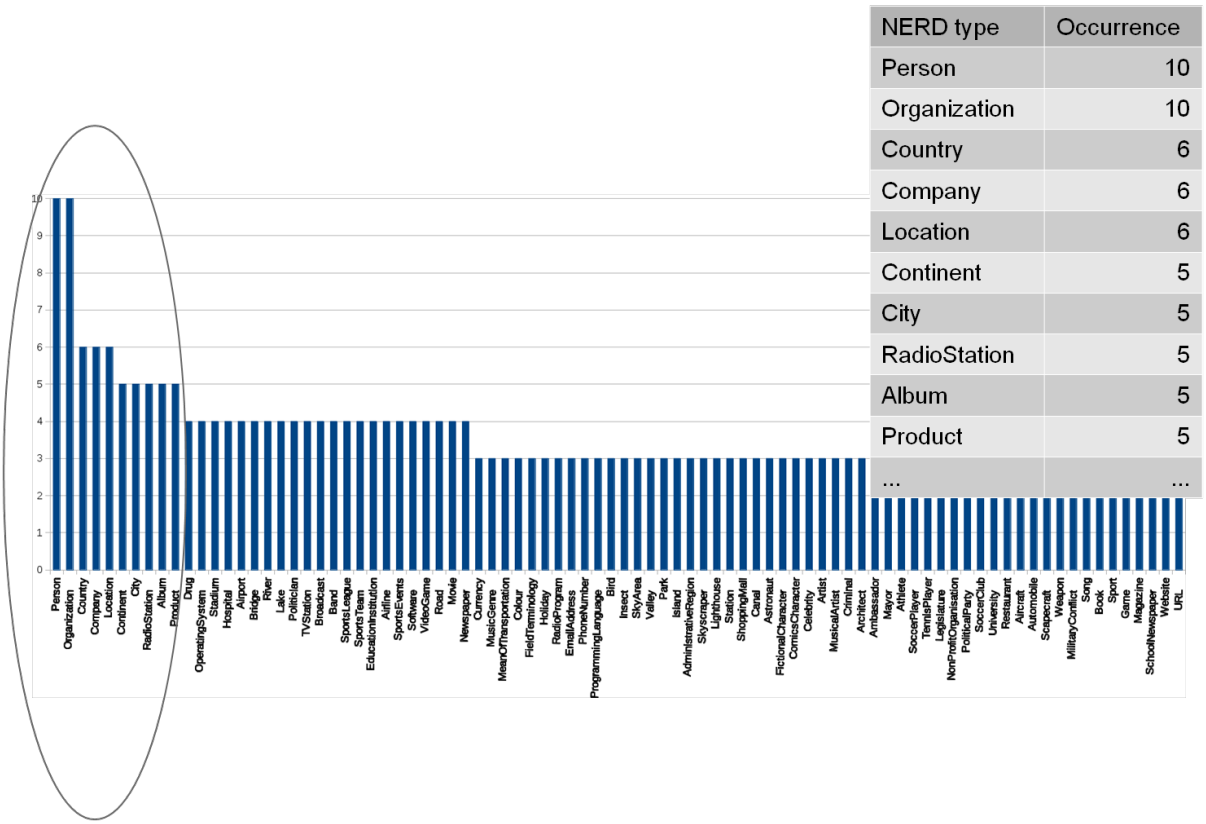
Although these tools share the same goal, they use different algorithms and different dictionaries which makes their comparison hard. We have developed the NERD ontology, a set of mappings established manually between the taxonomies of NE types. Concepts included in the NERD ontology are collected from different schema types: ontology (for DBpedia Spotlight, Lupedia, and Zemanta), lightweight taxonomy (for AlchemyAPI, Evri, and Yahoo!) or simple flat type lists (for Extractiv, OpenCalais, Saplo, and Wikimeta).

The NERD ontology tries to merge the linguistic community needs and the logician community ones: we developed a core set of axioms based on the Quaero schema [GRG+11] and we mapped similar concepts described in the other scheme. The selection of these concepts has been done considering the greatest common denominator among them. The concepts that do not appear in the NERD namespace are sub-classes of parents that end-up in the NERD ontology. This ontology is available at `http://nerd.eurecom.fr/ontology` (Figure 6).

To summarize, a concept is included in the NERD ontology as soon as there are at least two extractors that use it. The NERD ontology becomes a reference ontology for comparing the classification task of NE extractors. We show an example mapping among those extractors below: the `City` type is considered as being equivalent to `alchemy:City`, `dbpedia-owl:City`, `extractiv:CITY`, `opencalais:City`, `evri:City` while being more specific than `wikimeta:LOC` and `zemanta:location`.

```
nerd:City a rdfs:Class ;
  rdfs:subClassOf wikimeta:LOC ;
  rdfs:subClassOf zemanta:location ;
  owl:equivalentClass alchemy:City ;
  owl:equivalentClass dbpedia-owl:City ;
  owl:equivalentClass evri:City ;
  owl:equivalentClass extractiv:CITY ;
  owl:equivalentClass opencalais:City .
```

[57] http://jersey.java.net
[58] http://grizzly.java.net
[59] http://nerd.eurecom.fr/api/application.wadl



NERD type	Occurrence
Person	10
Organization	10
Country	6
Company	6
Location	6
Continent	5
City	5
RadioStation	5
Album	5
Product	5
...	...

Figure 6: NERD ontology: the long tail of common denominator between NER extractors taxonomies

4.2.4 NERD User Interface

The user interface[60] is developed in HTML/Javascript. Its goal is to provide a portal where researchers can find information about the NERD project, the NERD ontology, and common statistics of the supported extractors. Moreover, it provides a personalized space where a user can create a developer or a simple user account. For the former account type, a developer can navigate through a dashboard, see his profile details, browse some personal usage statistics and get a programmatic access to the NERD API via a NERD key. The simple user account enables to annotate any web documents via its URI. The raw text is first extracted from the web source and a user can select a particular extractor. After the extraction step, the user can judge the correctness of each field of the tuple *(NE, type, URI, relevant)*. This is an important process which gives to NERD human feedbacks with the main purpose of evaluating the quality of the extraction results collected by those tools [RT11a]. At the end of the evaluation, the user sends the results, through asynchronous calls, to the REST API engine in order to store them. This set of evaluations is further used to compute statistics about precision measures for each tool, with the goal to highlight strengths and weaknesses and to compare them [RT11b]. The comparison aggregates all the evaluations performed and, finally, the user is free to select one or more evaluations to see the metrics that are computed for each service in real time.

4.2.5 SemiTags

SemiTags is a web service for named entity recognition and disambiguation. It is intended to recognize named entities in unstructured texts and discover links to web based knowledge basis (namely Wikipedia and DBpedia). SemiTags works in two phases:

– Named Entity Recognition – The phrases corresponding to named entities are located in the text.

– Link Discovery – Local version of Wikipedia (corresponding to selected language) is searched for

[60] http://nerd.eurecom.fr

a suitable article describing entities located in the previous phase. The link to Wikipedia is then used to map the entity to the corresponding DBpedia resource (if available).

For named entity recognition in English and German SemiTags uses the state of the art Stanford Named Entity Recognizer [FGM05]. For Dutch we tested the use of OpenNLP[61] library trained on the CONLL-2002 [TKSDM03] datasets. However Stanford Named Entity Recognizer trained on the same dataset performs significantly better. The results of our evaluation are provided in Section 4.3.2.

For the second phase – Link Discovery – we consider the combination of the textual based approach introduced in [MJGSB11a] and structural based approach introduced in [MW08a] together with our structural based co-occurrence disambiguation. First of all, we generate the set possible candidates C to surface forms of named entities discovered in the text. If there is more than one candidate for a given surface form a disambiguation has to be performed.

Contrary to the approach presented in [MW08a] our structure based model does not compare similarities of individual entities. We are searching for the best combination of candidates for individual surface forms in the analyzed text. The whole text represents the context.

Consider for example the following sentence: *Michael Bloomberg is the mayor of New York*. Simple observation shows that the entity Michael Bloomberg (mayor of New York) co-occurs in the same paragraph in Wikipedia together with the correct entity New York City in United States much more often (88 times) than with the New York in England (0 times).

Because generating all candidate combinations is a very demanding task, we developed a heuristic that quantifies an impact of co-occurrences in the same paragraph.

We construct an incidence matrix I of the size $|C| \times |C|$ (where $|C|$ is the number of candidates), which represents a weighted graph. Weights are the co-occurrence measures and are assigned according to Equation 1.

$$d_{e_{i,s},e_{j,t}} = \begin{cases} 0 & \text{if } s = t \\ 0 & \text{if } i = j \\ |P_{e_{i,s},e_{j,t}}| & \text{if } i \neq j \text{AND } s \neq t \end{cases} \quad (1)$$

So the weight $|P_{e_{i,s},e_{j,t}}|$ (count of paragraphs, where e_i and e_j were mentioned together) is counted only in the case that the candidates represent a different entity $i \neq j$ and belong to a different surface form $s \neq t$, otherwise it is 0. Then we compute a score $e_{i,s}$ for each candidate as a sum of lines of the matrix representing the candidate (Equation 2).

$$e_{i,s} = \sum_{j=1}^{|C|} e_{i,j} \quad (2)$$

4.2.6 Targeted Hypernym Discovery (THD)

The Targeted Hypernym Discovery (THD) approach described here is based on the application of handcrafted lexico-syntactic patterns. Although lexico-syntactic patterns for hypernym discovery have been extensively studied since the seminal work [Hea92] was published in 1992, most research focused on the extraction of all word-hypernym pairs from the given generic free-text corpus.

4.2.6.1 Principle. Lexico-syntactic patterns were in the past primarily used on larger text corpora with the intent to discover all word-hypernym pairs in the collection. The extracted pairs were then used e.g. for taxonomy induction [SJN06] or ontology learning [CV05]. This effort was undermined by the relatively poor performance of lexico-syntactic patterns in the task of extracting *all* relations from a *generic* corpus. On this task, the state-of-the-art algorithm of Snow [SJN05] achieves an F-measure of 36 %.

However, applying lexico-syntactic patterns on a *suitable document* with the intent to extract *one hypernym* at a time can achieve F1 measure of 0.851 with precision 0.969 [LLM11]. In [LLM11], the suitable documents were Wikipedia entries for persons and the target of the discovery was the hypernym for the person covered by the article. Our THD algorithms is based on similar principles as [LLM11], but we do not limit is application to a certain entity type. The outline of the steps taken to find a hypernym for a given entity in our THD implementation is denoted in Alg. 1.[62][63]

[61] http://opennlp.apache.org/

[62]

[63]

Algorithm 1 Targeted Hypernym Discovery (getHypernym procedure)

Require: np – noun phrase representing the entity, $maxArticles$
Ensure: $hypernym$ – a hypernym for the entity

```
//if there are only n matching articles, n < maxArticles articles, get all
doc[] := get top maxArticles Wikipedia articles with title matching np
for i:=1 to |doc| do
  if doc[i] matches np then
    //extracts hypernym from the first paragraph of the Wikipedia article with Hearst patterns
    hypernym := extractHypernym(doc[i])
    if hypernym ≠ ∅ then
      return hypernym
    end if
  end if
end for
return ∅
```

The hypernym returned by Alg. 1 can be considered as the type of the entity represented by the noun phrase extracted from the text. In comparison with virtually all other entity classification algorithms, THD performs completely unsupervised classification: neither training examples nor the set of target classes is required. Should a classification to a user-defined set of classes be required, the entity type returned by THD can be used as an input for a more conventional entity classification algorithm.

The details relating to the THD algorithm as well as to the grammar used can be found in [KCN+08]. The advantage of the algorithm is that it can work against live Wikipedia, which fosters maximum freshness of the results. For performance reasons, an option to use an off-line copy of Wikipedia is also included.

4.2.6.2 Applying THD on German and Dutch. The design and evaluation of the THD algorithm was done so far with English as the target language. Completely porting THD to another language, requires:

1. the free availability of an encyclopedic resource (Wikipedia) for the given language,
2. availability of third party language processing tools (tokenizer and POS tagger) for the GATE framework[64],
3. devising grammar for entity extraction from the input text,
4. devising Hearst pattern extraction grammar for the language used.

Interestingly, once the named entities are extracted from the input text, they tend to be language independent. This extraction can be performed by other tool, such as Semitags, and the extracted entities can then be passed to THD. Wikipedia also contains redirects from different spelling variants of the named entity. For example, English Wikipedia contains a redirect from German “Brüssel” to “Brussels”. The hypernym returned by THD needs to be mapped to English DBpedia, which has been preliminarily accepted by the LinkedTV consortium as a component of the core ontology.

Of course, using Wikipedia of the particular language has its benefits, even for named entities. Local versions are smaller, but they are not subsets of English Wikipedia. Many named entities of local importance not present in the English Wikipedia are covered. However, use of non-English Wikipedia for THD would require the design of the extraction grammar for the particular language as well as the availability of other resources and processing tools as listed above. Also, an issue with mapping the non-English hypernyms to the English DBpedia may arise. As a conclusion, THD over English Wikipedia can be readily used in the project. We will however attempt to port THD to German and possibly Dutch.

4.2.6.3 NERD interface – NIF export. With the aim of achieving interoperability between THD and other NLP tools, we provide export of the processed results in the NIF format [HLA12]. The results from the *entity* and *hypernym extraction* together with information about their *resource representations* in DBpedia are translated into the NIF format and published as Linked Data.

[64] The THD is implemented on top of the GATE framework for text engineering (`http://gate.ac.uk`)

```
1  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  @prefix str: <http://nlp2rdf.lod2.eu/schema/string/>
3  @prefix dbpedia: <http://dbpedia.org/resource/>
4  @prefix sso: <http://nlp2rdf.lod2.eu/schema/sso/>
5  @prefix : <http://example.org/>
6  :offset_0_80_Diego+Armando+Maradona+Franco+is+from+Argentina.+Argentina+is+next+to+Chile.
7    rdf:type str:Context ;
8    str:isString "Diego Armando Maradona Franco is from Argentina. Argentina is next to Chile." ;
9  :offset_0_29_Diego+Armando+Maradona+Franco
10   rdf:type str:String ;
11   str:referenceContext :offset_0_80_Diego+Armando+Maradona+Franco+is+from+Argentina.+Argentina+is+
         next+to+Chile. ;
12   sso:oen dbpedia:Diego_Maradona ;
13   str:beginIndex "0" ;
14   str:endIndex "29" .
15   str:isString "Diego Armando Maradona Franco" ;
16 dbpedia:Diego_Maradona rdf:type dbpedia:Manager .
17
18 :offset_38_47_Argentina
19   rdf:type str:String ;
20   str:referenceContext :offset_0_80_Diego+Armando+Maradona+Franco+is+from+Argentina.+Argentina+is+
         next+to+Chile. ;
21   sso:oen dbpedia:Argentina_national_football_team ;
22   sso:oen dbpedia:Argentina ;
23   str:beginIndex "38" ;
24   str:endIndex "47" .
25   str:isString "Argentina" ;
26 dbpedia:Argentina rdf:type dbpedia:Country .
27
28 :offset_49_58_Argentina
29   rdf:type str:String ;
30   str:referenceContext :offset_0_80_Diego+Armando+Maradona+Franco+is+from+Argentina.+Argentina+is+
         next+to+Chile. ;
31   sso:oen dbpedia:Argentina_national_football_team ;
32   sso:oen dbpedia:Argentina ;
33   str:beginIndex "49" ;
34   str:endIndex "58" .
35   str:isString "Argentina" ;
36 dbpedia:Argentina rdf:type dbpedia:Country .
37
38 :offset_70_75_Chile
39   rdf:type str:String ;
40   str:referenceContext :offset_0_80_Diego+Armando+Maradona+Franco+is+from+Argentina.+Argentina+is+
         next+to+Chile. ;
41   sso:oen dbpedia:Chilean_peso ;
42   sso:oen dbpedia:Chile ;
43   str:beginIndex "70" ;
44   str:endIndex "75" .
45   str:isString "Chile" ;
46 dbpedia:Chilean_peso rdf:type dbpedia:Currency .
```

Figure 7: The excerpt of a NIF export.

The NIF export of our tool is based on the latest NIF specification and the recent feedback from the community.[65] Figure 7 presents an excerpt of a NIF export. The *input plain text* received for processing is considered as a *context* formalized using the OWL class **str:Context** (lines 7-9), within which entity candidates and their hypernyms need to be retrieved. Extracted entity candidates (i.e., string “Diego Armando Maradona Franco”) are treated as an offset-based string within the context resource (lines 10-16). With the help of the **sso:oen** property the underlying strings of the extracted entities get connected to their representation in DBpedia (line 13). Finally, discovered hypernyms for the entities are retrieved and DBpedia representation is discovered and attached to the entity (line 17).

4.2.7 Soft entity classification

So far, the description has focused on “crisp” classification - an input entity is typically assigned one class. This industry standard approach implies some limitations, with some specific for the LinkedTV use:

- the NER system is sometimes unsure which of the types is correct, however, just one type needs to be picked.
- in some cases, multiple types can be correct simultaneously. For example, the RBB entity can be simultaneously classified as nerd:MediaCompany and nerd:RadioNetwork.

[65] http://nlp2rdf.org/get-involved

– the result of NER in LinkedTV is used also for personalization: the type(s) of the entity present in the shot are aggregated to one feature vector, which is used by personalization algorithms. For this purpose, it is better to have a more robust entity representation (multiple types – possibly all in the ontology – with lower confidence), rather than a single type with non-negligible likelihood of being incorrect.

The above mentioned points can be addressed by providing soft (or sometimes referred to as "fuzzy") entity classification. Some tools described so far have the option to provide soft output. These systems include:

– DBpedia spotlight, included in the NERD platform, can be configured to return n-best candidates along with confidence levels.

– SCM algorithm [KCN+08], which uses THD algorithm to map entities to WordNet concepts, and then uses WordNet similarity measures to compute the similarity with each of the target classes. Target classes (concepts) are WordNet concepts.

– BOA algorithm [Kli10] is based on the Rocchio classifier applied on Wikipedia articles. Target classes (concepts) are Wikipedia articles.

The advantages provided by soft classification will be subject of further investigation, with the development focusing on the BOA and SCM algorithms.

4.2.8 Role of Semitags, BOA and THD within NERD

The NERD framework provides access to a range of third-party NER tools. At the time of writing, we observed that:

1. the support for German and Dutch is still limited,
2. many tools provide only crisp classification to one class without assigning confidence values (contrasting with WP4 requirements for confidence values and/or soft classification),
3. some of these tools provide only generic types while for personalization purposes, specific types are preferred,
4. these tools are third party services, sometimes commercial, and some of them might be interrupted.

The SemiTags, THD and BOA tools are implemented to complement the existing third-party serviced. The SemiTags tool was specifically developed to support German and Dutch, addressing the point (1). The BOA tool will provide soft entity classification to multiple entity types addressing the point (2). The THD tool outputs specific types for entities addressing the point (3). These tools are run on our hardware infrastructure mitigating the risk posed by the point (4).

4.3 NER Evaluation

4.3.1 NERD in the ETAPE Campaign

ETAPE is a project targeting the organization of evaluation campaigns in the field of automatic speech processing and natural language processing. Partially funded by the French National Research Agency (ANR), the project brings together national experts in the organization of such campaigns under the scientific leadership of the AFCP, the French-speaking Speech Communication Association, a regional branch of ISCA.

The ETAPE 2012 evaluation focuses on TV material with various level of spontaneous speech and multiple speaker speech. Apart from spontaneous speech, one of the originality of the ETAPE 2012 campaign is that it does not target any particular type of shows such as news, thus fostering the development of general purpose transcription systems for professional quality multimedia material. More precisely, the ETAPE 2012 data consists of 30 hours of radio and TV data from TV news, TV debates, TV amusements and Radio shows.

Several tasks are evaluated independently on the same dataset. Four tasks are considered in the ETAPE 2012 benchmark. For historical reasons, tasks belong to one of the three following categories: segmentation (S), transcription (T) and information extraction (E). The named entity task (E) consists in

detecting all direct mentions of named entities and in categorizing the entity type. The taxonomy follows the LIMSI Quaero definition as per the version 1.22 of the guide. Two conditions will be evaluated, detection on manual transcriptions and detection on ASR. At least one of the ASRs will be a rover. Entity types are organized in a hierarchical way (7 types and 32 sub-types):

1. Person: pers.ind (invidual person), pers.coll (collectivity of persons);
2. Location: administrative (loc.adm.town loc.adm.reg loc.adm.nat loc.adm.sup), physical (loc.phys.geo, loc.phys.hydro, loc.phys.astro);
3. Organization: org.ent (services), org.adm (administration);
4. Amount: quantity (with unit or general object), duration;
5. Time: date time.date.abs (absolute date), time.date.rel (date relative to the discourse), hour time.hour.abs, time.hour.rel ;
6. Production: prod.object (manufactury object), prod.art, prod.media, prod.fin (financial products), prod.soft (software), prod.award, prod.serv (transportation route), prod.doctr (doctrine), prod.rule (law);
7. Functions: func.ind (individual function), func.coll (collectivity of functions).

In order to participate in the campaign, we first built 426 axioms in the NERD ontology to the 32 concepts in the Quaero schema. The dataset being composed of French documents, we only consider the extractors Wikimeta, AlchemyAPI, Lupedia and OpenCalais. We developed a combined strategy of these 4 extractors which outperforms the performance of each individual extractor (Table 3).

	SLR	precision	recall	F-measure	%correct
AlchemyAPI	37,71%	47,95%	5,45%	9,68%	5,45%
Lupedia	39,49%	22,87%	1,56%	2,91%	1,56%
OpenCalais	37,47%	41,69%	3,53%	6,49%	3,53%
Wikimeta	36,67%	19,40%	4,25%	6,95%	4,25%
NERD combined	86,85%	35,31%	17,69%	23,44%	17,69%

Table 3: Performance comparison of the combined strategy of NERD with each individual extractor in the ETAPE campaign

The analysis per-type class highlights contrasted results: the class Person is generally well-detected while other category shows a very low recall. Interestingly, our approach performs equally on perfect transcriptions than on automatically transcribed texts which are generally noisy and grammatically incorrect. This proves that our approach is robust to non grammatically correct text since we are much less dependent on a specific learning corpora as traditionally performed by the other participants in this campaign. A much more thorough analysis of these results are being conducted at the moment for a journal publication.

4.3.2 SemiTags Evaluation

We tested two state of the art solutions for evaluating SemiTags: Stanford Named Entity Recognizer [FGM05] and OpenNLP[66] library. For the comparison, we used 10 manually annotated articles collected from the Dutch TV Show Tussen Kunst & Kitsch, which corresponds to the data sources for LinkedTV scenarios. Totally we identified 131 named entities in these texts.

Both tools were trained using the same CONLL-2002 [TKSDM03] datasets. In Figure 8 we show the results (overall precision and recall) of entity identification in the texts provided – in other words the ability of the tool to determine the exact position of the named entity.

Figure 9 shows the precision and recall of type determination in both tools. Note that this is the overall precision and recall of the identification and type determination. Thus, when an entity is not identified, it also results in an error in the type determination. Therefore, Figure 9 shows the results of the whole recognition process rather than just the type determination.

[66] http://opennlp.apache.org/

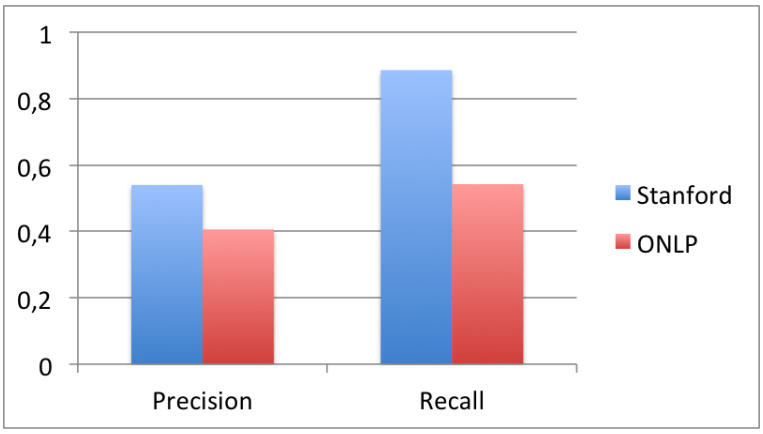


Figure 8: Precision and recall of entity identification using Stanford Named Entity Recognizer (Stanford) and OpenNLP library (ONLP).

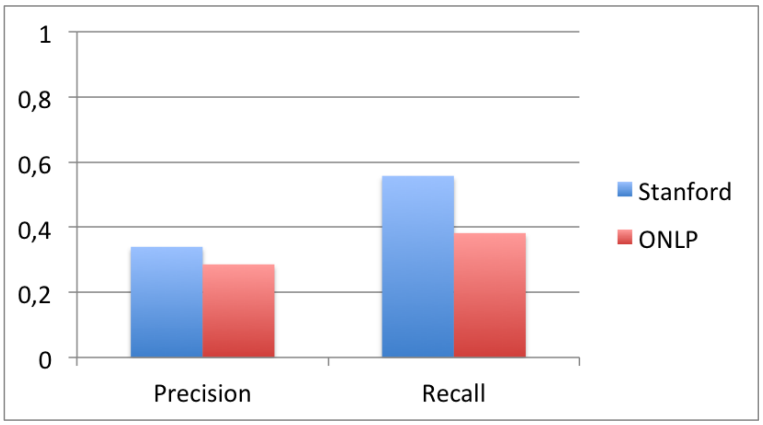


Figure 9: Precision and recall of type determination using Stanford Named Entity Recognizer (Stanford) and OpenNLP library (ONLP).

While results of the entity identification are acceptable, the performance of type determination is relatively poor. However, type determination will be validated in the next step of our disambiguation process using data found in our knowledge base indexed from Wikipedia. The Stanford Named Entity Recognizer outperforms the results of the OpenNLP library significantly. Thus, for further experiments we chose the tool provided by Stanford.

In our next experiment, we tested the Stanford Named Entity Recognizer trained on German texts with another manually annotated dataset of German articles. For testing the performance in German, we randomly collected 10 articles from RBB Online which is a white-listed data source. In German articles, we identified 121 entities. In Figure 10, precision and recall of entity identification in German and Dutch texts is shown.

Finally, Figure 11 shows precision and recall of the type determination of identified entities again in German and Dutch texts. The named entity recognition and type disambiguation processes have better results for German texts.

It is necessary to note that Tussen Kunst & Kitsch Web page in Dutch represents a much more difficult domain than German RBB News articles. The training data sets are focused on news domain. Therefore the recognition in this domain provides better results. Texts on Tussen Kunst & Kitsch Web page are also often partially structured (e.g. contain lists) and often do not contain whole sentences. For the recognizer, it is then very difficult to determine the boundaries of the named entities extracted.

As part of its interface, SemiTags provides a web service so that other tools can be connected to it and use its functionality. Currently, SemiTags is used within the NERD framework in this way. Apart from the web service, we developed also a demo with a web based user interface. In Figure 12, we show an example of the output of SemiTags running with a German article taken from RBB web site.

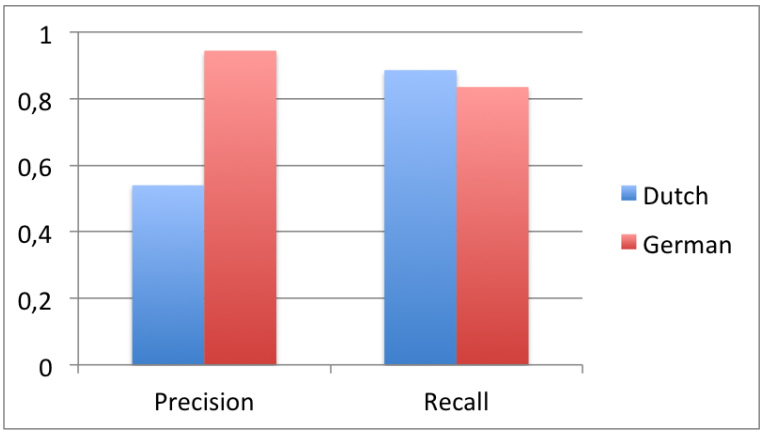


Figure 10: Precision and recall of entity identification – Named entity recognition for Dutch (using OpenNLP) and German (using Stanford Named Entity Recognizer).

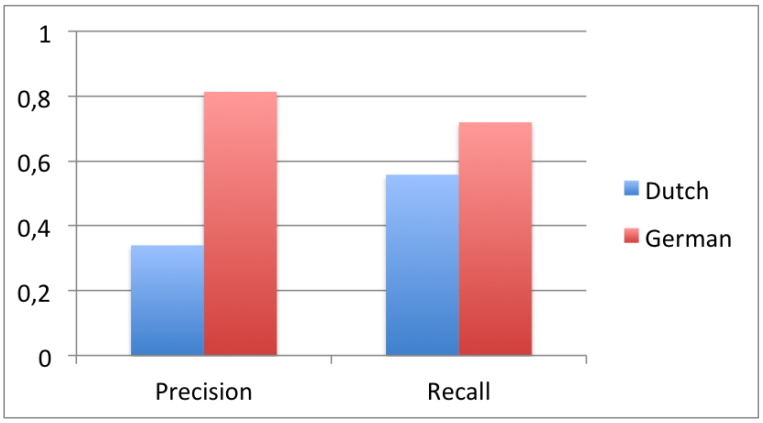


Figure 11: Precision and recall of type determination – Named entity recognition for Dutch (using OpenNLP) and German (using Stanford Named Entity Recognizer).

On the right side of the interface, we can see the actual results of named entity recognition (Identified Entities) and disambiguation (Disambiguated Named Entities). By entity recognition also a basic disambiguation is performed (we try to determine the basic type of a named entity (i.e. is it person, location, organization or other). After this phase, the more detailed disambiguation is performed which provides links to concrete Wikipedia articles describing a given entity. This can be seen in the lower right part of the interface. In the future, we plan to merge these results and provide the types rather based on the category of identified Wikipedia article.

4.3.3 THD Evaluation

In this section we present experiments with THD. The goal of the experiments is to compare our tool with several other entity extraction and classification tools. The THD algorithm is intended as complimentary to other NER algorithms. In our experience, its use is particularly beneficial on uncommon named entities, where other algorithms fail. To justify this, we have selected the CTC dataset (`http://ner.vse.cz/datasets/ctc`). The named entities subset of the CTC datasets consist of predominantly less common geographical names, which can be expected to appear in the "long tail" of the distribution of entities in the RBB use case. These names include e.g. "Korce" (Albania), "Velika planina" (Slovenia) or "Lenin". There are 101 named entities.

The experiments were run for the THD algorithm and three other SoA tools: DBpedia Spotlight (DB), Open Calais (OC) and Alchemy API (ALC). We used NERD to access these systems.

The experimental results presented in Figure 13 show that our tool produced almost consistently better results in all tasks than the other three tools. A qualitative comparison with DBpedia spotlight, the

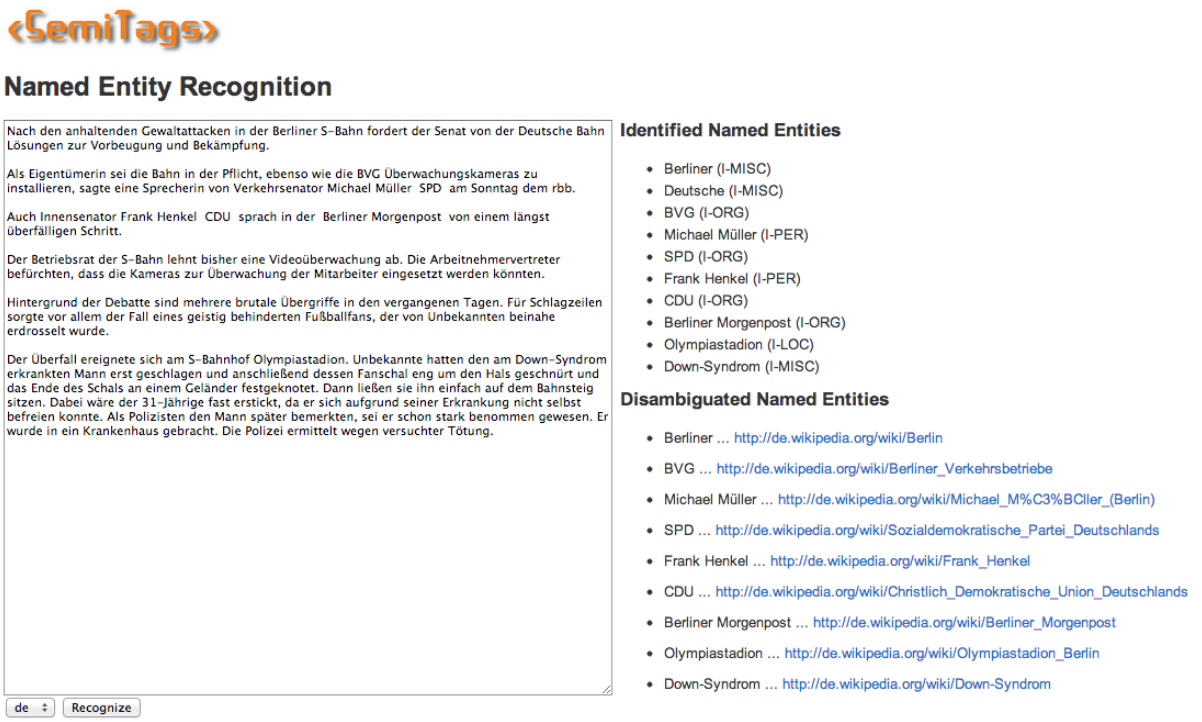


Figure 12: The example of SemiTags output available via its web based user interface. A randomly selected German article from RBB Online web site was used as the input text.

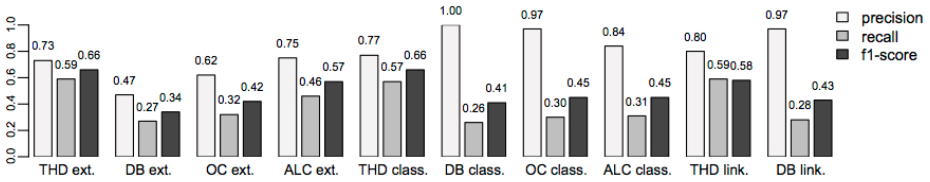


Figure 13: Evaluation of extraction, classification and linking of Named Entities

second best tool, is given in Table 4. A qualitative comparison between THD, DBpedia Spotlight and Semitags on several entities from the RBB dataset is present in Table 5.

The results in Table 5 confirm the complementary character of the tools. Additionally, both Table 4 and Table 5 demonstrate that THD gives more specific types.

Table 4: THD and DBpedia Spotlight comparison. Note that the links marked with * were not generated by Spotlight.

Entity	dataset	THD + Spotlight link	THD – type	Spotlight type
Vlore	CTC	http://dbpedia.org/page/Vlore_County	shore lines	dbo:Settlement, dbo:PopulatedPlace, dbo:Place
Korce	CTC	http://dbpedia.org/resource/Kor%C3%A7%C3%AB	city	dbo:Settlement, dbo:PopulatedPlace, dbo:Place, dbo:TopicalConcept
Lenin	CTC	http://dbpedia.org/resource/Vladimir_Lenin	communist	dbo:OfficeHolder, dbo:Person, dbo:TopicalConcept
Massandra	CTC	http://dbpedia.org/resource/Massandra*	townlet	NA

Table 5: THD, DBpedia and Semitags Qualitative comparison

Entity	dataset	THD – type	Spotlight	Semitags – de
Havelland	RBB	region	NA	NA
CSU	RBB	NA	NA	ORG
Magic Flute	RBB	Opera	NA	NA
Peter Schwenkow	RBB	NA	FIGURE	PERSON

5 Retrieving Additional Content from the Web – Requirements and Specification

Retrieving additional content from the Web relies particularly on entities recognized by WP2 as described in the previous section. Additional input data include the eXmaralda file, the subtitles, and the associated metadata if provided.

5.1 Functional Requirements and White Listed Resources

The additional content is retrieved mainly from a trusted list of resources, often called “white list”. Before being displayed in the LinkedTV media player, both linked content and concepts identified are filtered and presented through a collaboration between WP3 and WP4. The functional analysis depends on information provided for the particular scenarios from WP6 in two principal areas: list of white-listed Web sites and requirements for identification of relevant entities and additional content.

Before all, we see two requirements concerning the mining part:

- The mining process is not a real-time process, but is rather performed prior broadcasting happens. There are two reasons: 1) the provider must be able to review the associated links before releasing them to the public, 2) the computational requirements of some of the algorithms do not permit real-time execution.
- Linking of additional content will be made at the shot or scene level (not at the frame level): retrieving different pieces of content for each instant of the video would be of no use, and displaying of linked content to the user cannot be frame-based in order to be easily followed.

5.1.1 Types of Additional Content

Extraction of named entities will provide semantic concepts associated with the video. Given an entity in a shot and a relevant web resource related to this entity, the types of additional content include:

- link to that page (letting the user get the information (s)he is interested in from that page).
- related non-textual content (such as a video, image, etc.)
- factual information (e.g. birthday of a person entity)

Candidate content needs to be analyzed before retrieval, because the retrieval process is performed differently depending on the type of resource examined. Content can be described along two axes: the type of the web resource and the reliability of metadata (Table 5.1.1).

First, we distinguish diverse types of Web resources depending on their structure:

- Semantic Web Resources (LOD resources) – resources that provide information in a machine readable format as part of the Linked Data cloud. These resources are particularly important for obtaining additional information for named entities. They serve as a direct source of additional factual information that can be displayed to the user as well as source of data that are indirectly presented either in the form of mapped related content or results of API queries (e.g. position on the map displayed based on GPS coordinates obtained from LOD resources such as DBpedia or Geonames).
- Resources retrieved from public APIs – Resources that provide information via a custom API. These include sources of factual information or visual content mentioned in Sound and Vision scenarios (e.g. Europeana API or OpenImages.eu API) and also social networks such as Facebook included in RBB white list in the form of fan pages.
- Web pages – The majority of information remain in a plain text form, with some formatting marks often without a semantic meaning. Ordinary web pages constitute the majority of white listed pages in the RBB scenario. Sound and Vision scenarios list apart from structured resources also ordinary web pages like the home page of the Tussen Kunst & Kitsch TV show.
- Visual data – such as images and videos are another example of unstructured resources. When not analyzed, they do not hold a direct computer-interpretable semantic meaning.

	Semantic Web Resources	Public APIs	Web pages	Visual items
reliable metadata	a DBpedia resource	an object in a Europeana collection	a document in rbb online	a film clip discovered in OpenImages
ambiguous metadata	x	a Facebook post in one of the trusted profiles or group	a multimedia file with no associated data in a Web Page	some Youtube videos from a trusted chanel
no metadata associated	x	x	x	a random Youtube video

Table 6: Examples of different kinds of resources used for enrichment

It should be noted that semantic web resources and public APIs serve particularly as a mean to obtain a link to a web page, image or a video, which is displayable to the end user. Their second role is a retrieval of factual information.

Then, we distinguish three categories of data based on their origin and associated information:

- Resources coming from reliable sources that already have enough metadata describing their content are ready for retrieval.
- Resources with ambiguous metadata need to be verified before they can be added to the relevant document list (see 5.1.3). Verification involves further analysis of the content in order to determine the correctness of the metadata. For instance, the low level features of a video document will be processed to detect the concept associated with the metadata.
- Resources without any associated metadata need to be processed in order to extract information that will be queried during the retrieval process (see 5.1.3). For example, most Youtube videos fall into this category: they are uploaded with few or none (not reliable) associated data. For now on, we do not aim at indexing all Youtube content but we will only consider media coming from private archives (Youtube channels in the white list for example).

5.1.2 Retrieving Additional Content

We identified the following options for accessing and aggregating related content.

- Crawling – A methodical, automated manner to collect data from Web resources. A crawler systematically browses target Web resources and stores data locally - usually in a form of indexes to facilitate fast and easy search. As data are usually cached locally, a special attention should be paid to recrawling of particular resources in order to keep local data as fresh as possible. The majority of content are white-listed news web sites; e.g. RBB Aktuell[67] are suitable for crawling in order to download and index the content.
- Wrapper Based Extraction – Crawlers usually obtain data in a rather raw form. If some information can be derived from the structure of a Web page a wrapper based extraction may be useful. A wrapper enabling mining from a Web pages structure is generated, either manually (by hard coding extraction rules or automatically). There are two main approaches to automatic wrapper generation:
  - Wrapper induction [Kus97b] involves a significant effort in manual labeling of training examples.
  - Automated data extraction – an overview is given in [Liu07b].

  Wrapper based extraction techniques become useful in cases of pages, where we focus at some of its parts (e.g. containing video or audio). The video resources are sometimes widely represented in some white listed resources. The RBB scenarios include also linking to related pod casts. Wrappers may be used to identify and extract these media resources from the web page.

[67] `http://www.rbb-online.de/rbbaktuell/index.html`, RBB Online[68] or Deutsche Welle[69]

– Direct Querying – When a Web data source makes accessible an API or a SPARQL endpoint, it is possible to query the service directly. Visual content providers enable access to their services via APIs. This is also the case of YouTube and Flick considered by Sound and Vision scenarios.

– RDF data dumps – Most of Linked Data resources provide exports of their content in the form of RDF dumps (see Section 3.2.1). It is an effective way, how to load the data and cache them locally in order to speed up their search. Also Europeana provides RDF dumps apart from its API.

– Public Search Services – To locate general data on the Web, public search services such as Google Custom Search API or Yahoo! Boss may be used (see Section 3.3.1).

The techniques listed above enable the retrieval of candidate content. In some cases, this content may already be ranked according to its relevance[70]. Next phase in the process is to refine the search through text or content analysis with dedicated algorithms (see Section 5.2 and Section 5.3). The candidate content items will be saved to the triple store along with confidence scores and thus made available to WP3 and WP4 for subsequent analysis and processing.

The type of content determines to some extent the approach used to mine information from a particular resource. Given the characteristics of additional content, we use one or the other technique for retrieval. The workflow is constituted of different pipelines with a priority order (see Figure 2):

– If the content is a semantic web resource in the LOD cloud, it is queried using SPARQL. This is the most straightforward way to access information in a reliable fashion.

– Else, if the content is unstructured (or semi-structured) but can be accessed through a Web API, a query is made on the API by a custom-built client that enables to retrieve some content.

– Last option is for unstructured content from the white list that is not reachable through an API. This content is processed with crawlers, and if appropriate with wrappers. Textual content is saved to a full-text index, multimedia content is saved separately.

5.1.3 Further Processing and Storage of Additional Content

When crawling pages containing multimedia items, we retrieve those items along with surrounding textual information. The multimedia content is passed back to WP1 for processing. This process enables to access detailed information, and to index them accordingly. The WP1 architecture is crafted for analysis of videos; the various types of content that can be retrieved from the web (audio, images) corresponding to subtasks executed during video analysis. We will approach WP1 in order to have a lite version of its process specialized for audio and image content. The result of the processing is saved to the triple store.

Analyzing all the retrieved multimedia items can be considered as a brute force approach. It may reveal heavy and tedious work, and require a huge amount of resources. Future work includes attempts to use a radically different approach: we are considering focused processing, either based on the quantity of information already existing on the media (category of data as explained in section 5.1.1) or based on queries made for additional content. For instance, we could ask for only part of the analysis on a video, let's say the face analysis if the retrieval is focused on person entities.

Content retrieved in the last two pipelines (as mentioned in 5.1.2, i.e. unstructured content reachable or not through a Web API) may further be processed in order to refine the results list and return relevant content exclusively. The response format for the lists of content still needs to be defined in relation with WP3 and WP4, as it is an input for these work packages.

5.2 Retrieving Content by Text Analysis

The underlying data for retrieving additional content are annotations produced in WP1 (including results of automatic speech recognition) and meta data provided directly by content partners (e.g. subtitles, editor annotations). The task of this work package is to identify in these underlying texts and annotations real world concepts and map them to entities from Linked Open Data (LOD) cloud. This process is often called link discovery; named entity recognition is an essential part of it (see Section 2 for current state of the art techniques).

When LOD concepts or entities are identified, they are used to link the original video fragments with online content. We distinguish three types of content linked to a concept (see Section 5.1.1):

[70] Public search services will return already ranked content, while crawling will not.

– Structured Resources – Direct sources of structured information about a concept (e.g. place or date of birth of a person).

– Semi-Structured and Unstructured resources – Mostly ordinary Web pages that possibly contain various media. By retrieving additional content from these resources, we mean identifying either whole Web pages related to a video or some part of it like multimedia (e.g. videos, audio and images).

Possible sources of data on the Web are covered in Section 3.2.1. The main resource of structured data is the LOD cloud, where the starting point is DBpedia [ABK+07]. Recognized named entities are linked to DBpedia concepts and their identifiers are used to crawl additional information from other LOD resources. The majority of online content comes from white listed resources approved by content partners. For example, if the video is a news show, we aim at linking particular spots to appropriate news from the portals white listed by the content provider.

The way of consumption of individual Web pages depends on their character. Ordinary Web pages (like news servers) can be crawled (Section 3.1.1) and indexed locally in order to support fast search for related concepts. Figure 14 shows the process of crawling and indexing web pages. Crawling starts with the list of whitelisted resources to be crawled from the web. Individual web pages are downloaded and further processed in order to identify basic structured information about the web page. This include title and description of the web page, as well as main text blocks and media content contained in the web page.

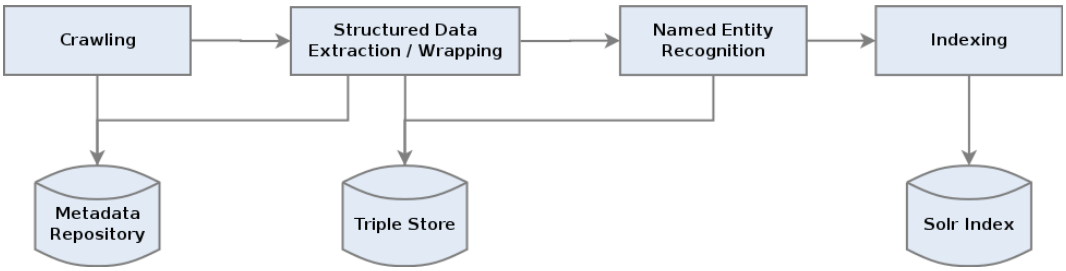


Figure 14: Web resources processing pipeline

The crawler maintains a local metadata repository, where the pieces of operational information are stored. These include time of the last download of the particular resource, checksum to track changes and identified structure of the web page. From solutions covered in Section 3.1.1 we select state of the art Nutch crawler, which is probably the most popular open source crawler that provides the necessary level of extensibility thanks to its plugin architecture.

Structured information extracted from the web pages is also pushed further in the processing pipeline. Textual representation of the crawled web page is analysed by named entity recognition and concrete LOD concepts are identified in extracted texts. Extracted texts together with extracted pieces of structured information are stored in the central triple store for further processing by other WPs. Finally, the results of crawling are indexed in Solr index to support faster querying and retrieval of related content for videos.

The process of retrieval of related content is displayed in Figure 15. The process starts with the video representation produced in WP1. Here, we consider among all the textual representation of the video and its metadata (i.e. subtitles, results of automatic speech recognition, manual video annotations, identified keywords). These results are processed by Named Entity Recognition tools (SemiTags, THD and others included in NERD framework). Identified named entities are disambiguated and mapped to LOD concepts – identifiers in the form of URIs are assigned to the entities. Follows the process of enrichment of named entities – the URIs are used to query LOD data sources to obtain additional information about identified entities (e.g. the place and date of birth in case of the name of a person, exact location in case of the name of a place etc.).

Identified concepts together with textual representation of the video are used to query the Solr index maintained during crawling process. Concrete tuning of the index is the subject of our future evaluation. The basic approach is to use named entities identified in the video and find crawled resources with the biggest overlap in the contained entities. In other words, entities extracted from videos are used as queries instead of plain keywords. Similar approach is used in KIM platform [PKK+03] to identify related news.

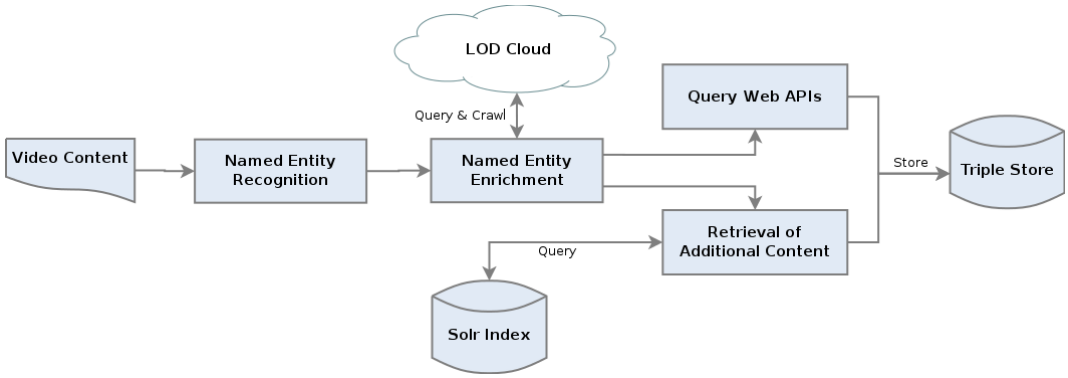


Figure 15: Web resources processing pipeline

In order to retrieve general content not necessarily limited to white listed resources, public search services and public APIs (Section 3.3) are a great source of information. For example services like Google Images or Ookaboo API[71] help to locate illustrative images anywhere on the Web. Finally, the results of this processing pipeline are again stored in central triplestore.

5.3 Retrieving Content by Visual Analysis

Retrieving multimedia files such as images or videos, in relation with the displayed video, is a key part of LinkedTV. Our goal is to search for additional content based on textual data mainly, but visual features of the images/videos can be of great use to complement retrieval by text analysis.

As explained in 3.4, retrieval by content analysis is based on the results of low-level features analysis (performed by WP1). WP2 focuses on the retrieval process based on the results of this analysis. The most appropriate query here is query by example, as it relies on visual features present in the broadcast video. It is made on different basis:

- Object re-detection and face detection and recognition are performed in WP1. Those methods aim at clustering similar faces and objects. WP1 gives us access to a set of bounding boxes in video frames that contain objects of interest or faces. Those bounding boxes can be used for matching objects (or faces) to objects (or faces) in additional videos. This may be done in complement to retrieval by keyword. For instance, in the cultural heritage scenario from Sound and Vision, art objects are presented to the viewers by experts during a show, “Tussen Kunst & Kitsch”. The scenario is focused on enriching objects, locations and people in the programme with high quality related items. Hence, a search can be made on objects extracted from the video to match them to objects from Europeana (one of the white-listed resources for this scenario) depending on visual similarity (computing features such as SURF, part of the WP1 process).

- Retrieval from global features of images is also a possible option. It would lead to retrieving content that has a similar visual layout as the one presented as query. This is interesting when looking for a landscape for example or a similarity between scenes. Again, visual features computing is part of WP1 process, so retrieval by content analysis would be a joint process of WP1 and WP2.

As said earlier in the scenario analysis, visual features can be used as well to perform mining in order to extract clusters of similar objects.

Relevance feedback, which is widely used in retrieval by content as described in section 3.4, is not applicable at retrieval time in the scope of the LinkedTV project, because the user is not involved in the retrieval process. Indeed, this process is done prior to broadcasting and at that time interaction with the viewers is not possible. Links to additional content are displayed to the user watching television; they are extracted from a list of relevant content filtered by the personalization and presentation layers. Work on retrieval by content analysis will be further described in the coming deliverables as it is still work in progress. It is in the scope of year 2 work and involves a loop with WP1 that performs content analysis based on low-level features.

[71] http://ookaboo.com/

5.4 Specific Requirements of Individual Scenarios and White Lists

RBB and Sound and Vision provided white lists of resources that are trusted potential sources of additional content. Only the white listed resources will be processed within WP2. All potential relevant source of information should therefore be added to the white list in order to be taken into account: for instance, if a provider considers Wikipedia as a trusted resource whose information is worth reading, it should be included in the white list. We expect the content providers to add new resources and expand the description of the existing resources in line with the taxonomy provided below:

- The types of resources appearing in the white-list determine the suitable techniques for retrieval of additional content. In this respect, we can distinguish between the following three types of Web sites:
  - Web sites exposing programmatic API.
  - Highly structured Web sites.
  - Web sites with prevalent textual content.
- Types of resources within the Web site to link to:
  - Web pages (URL's),
  - Position within Web page using existing anchors present on the Web page,
  - Fragments of Web pages such as relevant paragraphs of free text[72]
  - Images
  - Video content
- Recency - how frequently should the crawling be performed to assure relevant results. This is linked with the availability of an up-to-date site map for each resource, which provides a list of change timestamps for individual Web pages. Also, some content may be deleted from a website. For instance, it is not possible to link to content on rbb website after 7 days, which means that the crawling must store timestamps and apply a "visible" or "not visible" tag for each resource.
- The size of individual Web sites in terms of the number of Web pages and number of changes per time period.

Scenarios as described by WP6 define the kind of resources to be displayed and impose additional requirements on the outcome of the mining process. The main sources of additional content in the video are related to the following: what (event, object), who (person), when (date, event), where (place). The input of the process are the linked entities found by WP2 for the video fragments; the corresponding URIs give access to structured information that can be displayed to the user.

5.4.1 Sound and Vision scenario

The Sound and Vision institute provides a documentary scenario in the field of cultural heritage. The scenario is based on the Dutch show "Tussen Kunst & Kitsch", and is focused on enriching the broadcast video with added content on objects, locations and people appearing in the programme.

Symbol example. One of the episodes from the Tussen Kunst & Kitsch show display a golden box in which the Chi Ro symbol has been incorporated. The scenario depicts a woman, Rita, who wants to know more about this symbol. The first step for enriching content about it is to identify the concept "chi rho symbol" in the shots. A DBpedia entry provides additional links. We can propose the following additional content:

- an article from Wikipedia, that contains description and pictures about the symbol.
- Europeana enables to see different object with the same symbol
- related content such as other related symbols based on the categories listed in DBpedia: links to Christian symbols, Roman-era Greek inscriptions, early Christian inscriptions

[72] It should be noted that if this requirement is imposed, it is unclear how to address a fragment of free text (fragment of a Web page). For third party Web pages, it is not possible to insert custom tags (e.g. HTTP anchors).

Figure 16: Frames extracted from the Tussen Kunst & Kitsch show that highlight found entities: the Chi Rho symbol and the Delftware

Style example. Another show presents a Delftware plate made in 1670. In the related segments, we can identify the concepts "Delftware" linked to the URI `http://dbpedia.org/page/Delftware`. Delftware refers to pottery made in the Netherlands from the 16th century, around the city of Delft. The proposed content could be:

- archive video from 1976 found on the Open Images website (openbeelden) that presents pottery making in the town
- some Delftware examples can be displayed, with images taken from the Amsterdam Museum or Flickrwrapper (`http://www4.wiwiss.fu-berlin.de/flickrwrappr/`).
- books from the Worldcat catalog related to this item.

Object example. Objects are very important in S+V scenarios, as they are the main focus of the seed shows. Diverse art objects (paintings, vases, sculptures, etc) are displayed for the viewers. Mining for similar objects is an important part of the scenarios. Displaying links to similar objects in Europeana or museums (Amsterdam museum is in the white list and is available as LOD) is of particular interest. Linked entities such as "Delftware'" can help to retrieve similar content. Visual analysis may be interesting here to find visually similar objects in the media from the white lists. Last, we can also enrich the videos by browsing through the archives of the show to mine similarities in presented objects. Thus, we could link to videos fragments in the archive of the show displaying visually similar objects, not only corresponding to the same style (this is done with the entity identification), but similar according to shape, color, interest points or other matching descriptors. Figure 17 shows clusters of similar objects that could be browsed through by the viewer.

5.4.2 RBB's scenario

RBB scenario refers to local news. It uses episodes of its daily local news program "RBB Aktuell" as the seed videos.

BER airport example. A news report in one of the programmes refers to the opening of the Berlin airport. Therefore, the concept "Airport BER" is linked to the given fragment, and related information is retrieved concerning:

- State of things. A link could be made to the web site `http://www.rbb-online.de/themen/flughafen-ber/flughafen_ber/index.html` that gives the most recent news on the airport. We can either directly display the link to the whole website as it focuses on the airport construction, or crawl the website in order to link to specific video content.
- Politics, and opinions on the event. RBB online contains multiple reports related to the airport. Suggestions of additional content can include `http://www.rbb-online.de/themen/flughafen-ber/flughafen_ber/bildergalerien/reaktion_auf_termin.html` for politicians opinions and `http://www.rbb-online.de/abendschau/archiv/archiv.media.!etc!medialib!rbb!rbb!abendschau!dossier!abendschau_20120509_situation.html` for people opinions.

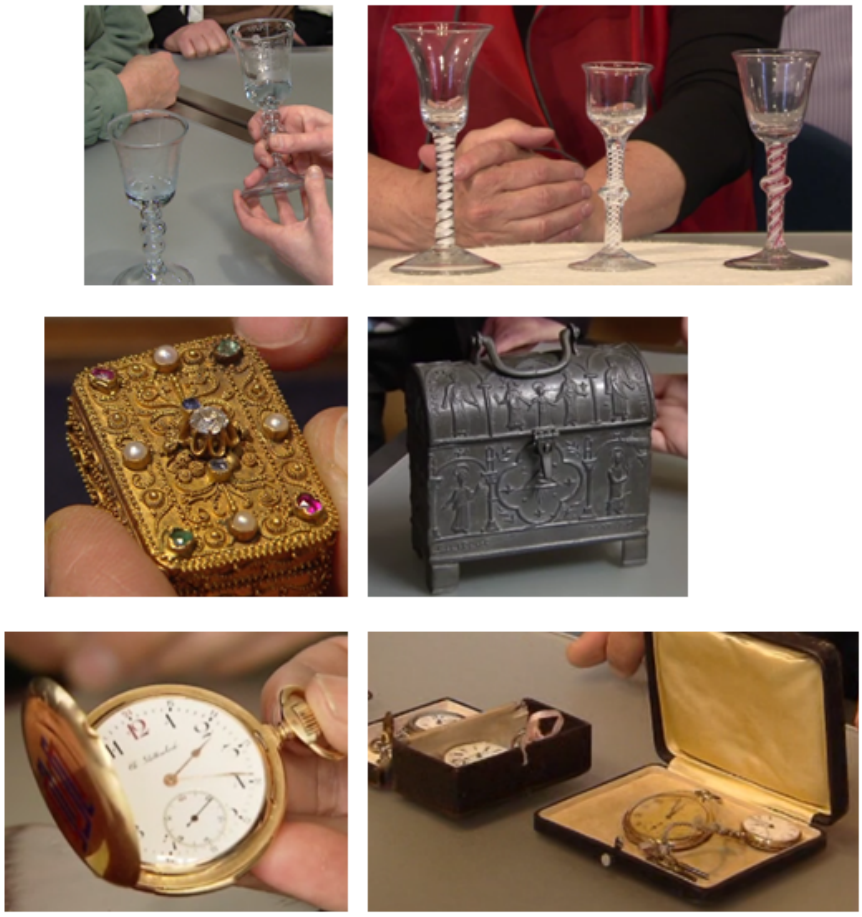


Figure 17: Each row represents objects that could be clustered according to visual similarities (shape, salient points, etc)

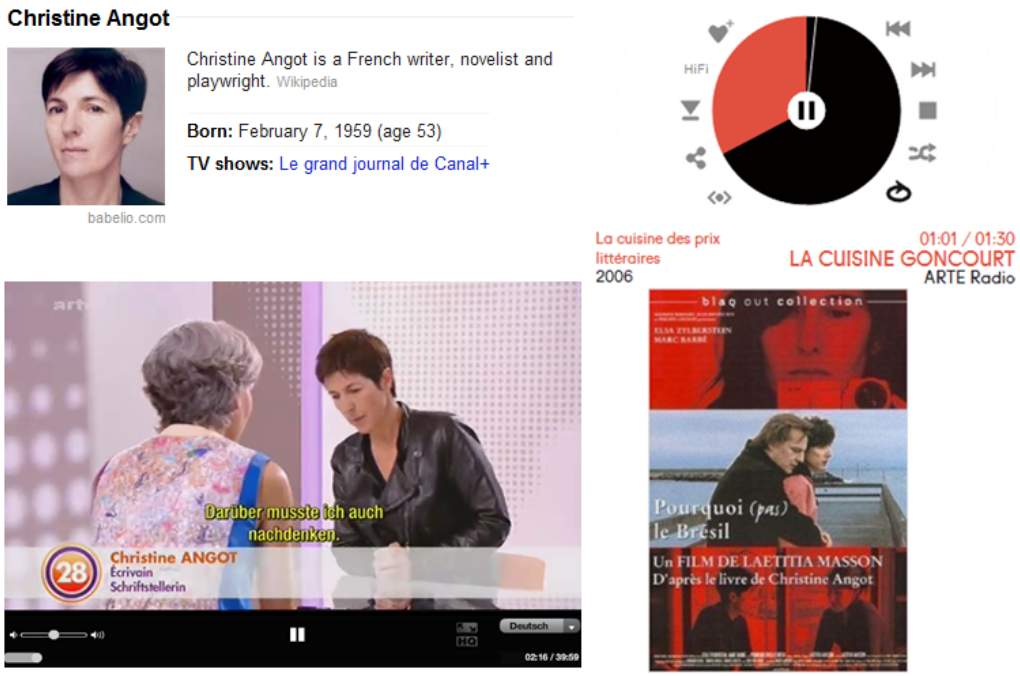


Figure 18: Candidate content for the entity "Christine Angot": factual information, an audio extract, a poster from a movie and a video interview from a show

- Architecture. A video giving an animated model of the airport can be found in the ZDF website[73].
- Geography. The airport can be situated on a map with GoogleMaps or OpenStreetMaps.
- Other. NDR website hosts the radio-comedy "Fruhstuck bei Stefanie". One episodes, Vertudelt, deals with the topic of the airport `http://www.ndr.de/ndr2/start/fruehstueck_bei_stefanie/videos/fbs1061.html`. As part of the white-list, this content has been crawled and indexed. It can be retrieved by a text-search on the associated textual data (mainly the description).

Person example. In the scenarios, people are also a key element; users are interested in additional information about the people they see. We take the example of a program from RBB featuring Christine Angot (`http://dbpedia.org/page/Christine_Angot`). When looking at RBB's white list, we can find diverse relevant information:

- A short biography can be found on Wikipedia, along with books and filmography.
- 28 minutes show from Arte Video[74] where a large part is about the French author. Here, analysis of the video enables to link directly to the relevant fragment.
- La cuisine Goncourt from Arte Audio,[75] (1min30).
- IMDB cites numerous shows she appears in. Also, the poster of "Pourquoi (pas) le Bresil", a movie inspired from her novel, can be displayed to the viewer.

Place example. Places where event occurs are another source of interest. In the scenario, Peter is watching a report on a reading in an old palace, at Schloss Neuhardenberg. The user is interested in this place and wants to know more about it, its exact location in particular. The following content can then be proposed to link with places:

[73] `http://www.zdf.de/ZDFmediathek/beitrag/video/1727052/Animation-zum-neuen-Hauptstadtflughafen`
[74] `http://videos.arte.tv/fr/videos/28-minutes--6905576.html`
[75] `http://www.arteradio.com/son/23893/la_cuisine_goncourt/`

- Displaying location on a map using OpenStreet Maps or Google Maps.
- Geographic information and factual informations. DBpedia can help with this.
- Touristic informations, videos and images
- Local information such as local news report (that can be extracted from RBB's website and from its partners')
- Events. In this place, diverse lectures and concerts are mentioned in article on `http://www.radioeins.de/` that can be accessed through crawling. Also, a video report on an exhibition on the history of garden art can be found on the web page`http://www.rbb-online.de/fernsehen/medienpartnerschaften/stiftung_schloss_neuhardenberg1.html`. This video can be retrieved after a crawling phase followed by scraping to extract the video.
- Venues, in Foursquare for example.

Several concepts in a segment. RBB scenario highlights the case where more than one entity is found in a video segment. For example, a segment can contain an entity concerning a place and another one concerning a person. We can add the extra content in two different ways:

- Either additional content is retrieved for each concept separately, hence there will be a list of potential content for display in relation with each entity.
- Or both entities can be the seeds for retrieval of additional content, i.e. we are interested in the relationship between the two, and we look for content where they overlap. For instance, in the news report describing a reading at Schloss Neuhardenberg, both the actor doing the reading, Klaus Maria Brandauer (`http://dbpedia.org/page/Klaus_Maria_Brandauer`) and the palace are recognized entities. The direction of the retrieval process could be the lecture of other actors in the same place, or movies he appears in, that happen in a castle.

5.4.3 Supporting the requirements

The previous section lists multiple requirements imposed by the scenarios on the functionality provided by WP2. Multiple requirements map to the same functionality. For example, getting a Wikipedia description for the Chi-Rho symbol and for Delftware require the same functionality. While the developments of some tools already started during year 1, other tools are only envisaged, and their provision and functionality may be subject to slight change depending on the progress of the research and refinement of the scenario requirements.

6 Conclusion and Future Work

In this deliverable, we already described the first software results for entity recognition. The consortium has developed mature tools for performing named entity recognition and disambiguation for Dutch, German and English. The NERD framework provides an umbrella over multiple third-party services while innovating in proposing novel combined strategy that outperforms single NER extractor. As a future work in this area, we will perform more rigorous evaluation of the THD algorithm on German and Dutch dataset with the possibility of porting this tool to one or both languages as an option. Work will be also directed at complementing the rule-based THD algorithm with a semi-supervised statistical BOA algorithm, which also uses Wikipedia as the knowledge source, but is language independent due to its statistical nature.

The work on software support for additional content retrieval is starting. Based on the analysis of representative requirements expressed in D6.1, we have described in the section 5.4 the required functionalities. Some of these requirements are satisfied with tools developed in year 1, the existence of others requires new tools to be drafted. The techniques planned for linking to media resources are: a) using SPARQL queries for semantic web resources, b) developing custom wrappers for Web APIs, c) developing a web retrieval module for crawling, indexing and analyzing web pages from the white lists. An important issue to be analyzed in the future, pertaining to communication between WP2 and WP4, is the representation of additional content that will be employed for filtering the results. A refinement based on a more thorough analysis of the content-providers approved white-lists will be carried out in year 2.

References

[ABK+07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe CudrÃł-Mauroux, editors, *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer Berlin / Heidelberg, 2007.

[ACGM+01] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan. Searching the Web. *ACM Trans. Internet Technol.*, 1(1):2–43, August 2001.

[AGM03] Arvind Arasu and Hector Garcia-Molina. Extracting structured data from Web pages. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 337–348, New York, NY, USA, 2003. ACM.

[AM02] E. Alfonseca and S. Manandhar. An unsupervised method for General Named Entity Recognition and Automated Concept Discovery. In *Poceedings of the First International Conference on General WordNet*, Mysore, India, 2002.

[AM03] Masayuki Asahara and Yuji Matsumoto. Japanese Named Entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 8–15, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[any12] Any23 – Anything to Triples. http://any23.org/, 2012.

[APR+10] Manuel Álvarez, Alberto Pan, Juan Raposo, Fernando Bellas, and Fidel Cacheda. Finding and Extracting Data Records from Web Pages. *Journal of Signal Processing Systems*, 59:123–137, 2010.

[AT] Javad Akbari Torkestani. An adaptive focused Web crawling algorithm based on learning automata. *Applied Intelligence*, pages 1–16.

[ATDE09] Eytan Adar, Jaime Teevan, Susan T. Dumais, and Jonathan L. Elsas. The web changes everything: understanding the dynamics of web content. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 282–291, New York, NY, USA, 2009. ACM.

[BCD05] Roberto Basili, Marco Cammisa, and Emanuale Donati. RitroveRAI: A Web Application for Semantic Indexing and Hyperlinking of Multimedia News. In Yolanda Gil, Enrico Motta, V. Benjamins, and Mark Musen, editors, *The Semantic Web âĂŞ ISWC 2005*, volume 3729 of *Lecture Notes in Computer Science*, pages 97–111. Springer Berlin / Heidelberg, 2005.

[BCSV04] Paolo Boldi, Bruno Codenotti, Massimo Santini, and Sebastiano Vigna. UbiCrawler: a scalable fully distributed Web crawler. *Software: Practice and Experience*, 34(8):711–726, 2004.

[BEP+08] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1247–1250, New York, NY, USA, 2008. ACM.

[Ber01] Michael K. Bergman. The Deep Web: Surfacing Hidden Value. *JEP: The Journal of Electronic Publishing*, 7(1), 2001.

[BFF04] Luciano Barbosa, Juliana Freire, and Juliana Freire. Siphoning hidden-web data through keyword-based interfaces. In *SBBD 2004 – 19th Brazilian symposium on databases*, pages 309–321, 2004.

[BFG01] Robert Baumgartner, Sergio Flesca, and Georg Gottlob. Visual Web Information Extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 119–128, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[BHBL09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[BKvH02] Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In Ian Horrocks and James Hendler, editors, *The Semantic Web ? ISWC 2002*, volume 2342 of *Lecture Notes in Computer Science*, pages 54–68. Springer Berlin / Heidelberg, 2002.

[BL06] Tim Berners-Lee. *W3C*, 2009(09/20):7, 2006.

[BLK+09] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154 – 165, 2009.

[BLPP01] David Buttler, Ling Liu, Calton Pu, and Calton Pu. A Fully Automated Object Extraction System for the World Wide Web. 2001.

[BMSW97] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, ANLC '97, pages 194–201, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107 – 117, 1998.

[BPM09] Sotiris Batsakis, Euripides G.M. Petrakis, and Evangelos Milios. Improving the performance of focused web crawlers. *Data & Knowledge Engineering*, 68(10):1001 – 1013, 2009.

[BPP06] Razvan Bunescu, Marius Pasca, and Marius Pasca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06), Trento, Italy*, pages 9–16, 2006.

[BPV94] R. Basili, M. T. Pazienza, and P. Velardi. A "not-so-shallow" parser for collocational analysis. In *Proceedings of the 15th conference on Computational linguistics - Volume 1*, COLING '94, pages 447–453, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[Bri99] Sergey Brin. Extracting Patterns and Relations from the World Wide Web. In Paolo Atzeni, Alberto Mendelzon, and Giansalvatore Mecca, editors, *The World Wide Web and Databases*, volume 1590 of *Lecture Notes in Computer Science*, pages 172–183. Springer Berlin / Heidelberg, 1999.

[BSAG98] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. NYU: Description of the MENE Named Entity System as Used in MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7*, 1998.

[Bur97] Mike Burner. Crawling towards eternity: Building an archive of the World Wide Web. *Web Techniques Mag.*, 2(5), 1997.

[BYRN11] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.

[CDD+04] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, WWW Alt. '04, pages 74–83, New York, NY, USA, 2004. ACM.

[CDK+99] Soumen Chakrabarti, Byron E. Dom, S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, Jon Kleinberg, and Jon Kleinberg. Mining the Web's Link Structure. pages 60–67, 1999.

[CGM99] J. Cho and H. Garcia-Molina. The Evolution of the Web and Implications for an Incremental Crawler. Technical Report 1999-22, Stanford InfoLab, 1999.

[CGMP98] Junghoo Cho, Hector Garcia-Molina, and Lawrence Page. Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30(1âĂŞ7):161 – 172, 1998.

[Cha02] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 1st edition, October 2002.

[Cha03] Soumen Chakrabarti. Mining the Web: Discovering Knowledge from Hypertext Data. 2003.

[CHJ02] William W. Cohen, Matthew Hurst, and Lee S. Jensen. A flexible learning system for wrapping tables and lists in HTML documents. In *Proceedings of the 11th international conference on World Wide Web*, WWW '02, pages 232–241, New York, NY, USA, 2002. ACM.

[CHSS08] Chaitanya Chemudugunta, America Holloway, Padhraic Smyth, and Mark Steyvers. Modeling Documents by Combining Semantic Concepts with Unsupervised Statistical Learning. In Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 229–244. Springer Berlin / Heidelberg, 2008.

[CL01] Chia-Hui Chang and Shao-Chen Lui. IEPAD: information extraction based on pattern discovery. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 681–688, New York, NY, USA, 2001. ACM.

[CMMM01] Valter Crescenzi, Giansalvatore Mecca, Paolo Merialdo, and Paolo Merialdo. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In *Proceedings of 27th International Conference on Very Large Data Bases*, pages 109–118, 2001.

[CQQ09] Gong Cheng, Yuzhong Qu, and Yuzhong Qu. Searching Linked Objects with Falcons: Approach, Implementation and Evaluation. pages 49–70, 2009.

[CSS99] Michael Collins, Yoram Singer, and Yoram Singer. Unsupervised Models for Named Entity Classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

[CV01] Alessandro Cucchiarelli and Paola Velardi. Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Comput. Linguist.*, 27(1):123–131, March 2001.

[CV05] Philipp Cimiano and Johanna Völker. Text2Onto: a framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th international conference on Natural Language Processing and Information Systems*, NLDB'05, pages 227–238, Berlin, Heidelberg, 2005. Springer-Verlag.

[CvdBD99] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks*, 31(11-16):1623 – 1640, 1999.

[dbw06] DBWorld. http://www.cs.wisc.edu/dbworld/, 2006.

[DCL+00] M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles, and M. Gori. Focused crawling using context graphs. In *In 26th International Conference on Very Large Databases, VLDB 2000*, pages 527–534, 2000.

[Del09] Renaud Delbru. SIREn: Entity Retrieval System for the Web of Data. In *Proceedings of the 3rd Symposium on Future Directions in Information Access (FDIA)*, University of Padua, Italy, 2009.

[DFJ+04] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal Doshi, and Joel Sachs. Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 652–659, New York, NY, USA, 2004. ACM.

[DPMM01] Melania Degeratu, Gautam Pant, Filippo Menczer, and Filippo Menczer. Latency-dependent fitness in evolutionary multithreaded Web agents. 2001.

[EJN99] D. W. Embley, Y. Jiang, and Y.-K. Ng. Record-boundary discovery in Web documents. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, SIGMOD '99, pages 467–478, New York, NY, USA, 1999. ACM.

[Etz05] Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 165(1):91 – 134, 2005.

[Eva03] Richard Evans. A framework for named entity recognition in the open domain. In *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, 2003.

[FGM05] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[FMNW04] Dennis Fetterly, Mark Manasse, Marc Najork, and Janet L. Wiener. A large-scale study of the evolution of Web pages. *Software: Practice and Experience*, 34(2):213–237, 2004.

[FP10] Manaal Faruqui and Sebastian Padó. Training and evaluating a german named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, Saarbrücken, Germany, 2010.

[FSN+95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the QBIC system. *Computer*, 28(9):23 –32, sep 1995.

[FT02] Roy T. Fielding and Richard N. Taylor. Principled design of the modern web architecture. *ACM Transaction Interneternet Technology*, 2:115–150, May 2002.

[GBH+07] Wolfgang Gatterbauer, Paul Bohunsky, Marcus Herzog, Bernhard Krüpl, and Bernhard Pollak. Towards domain-independent information extraction from web tables. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 71–80, New York, NY, USA, 2007. ACM.

[geo12] GEOnet Names Server. http://earth-info.nga.mil/gns/html/, 2012.

[GGS05] Alfio Gliozzo, Claudio Giuliano, and Carlo Strapparava. Domain kernels for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 403–410, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[GNI12] Geographic Names Information System - GNIS. http://nhd.usgs.gov/gnis.html, 2012.

[GNP+09] Daniel Gruhl, Meena Nagarajan, Jan Pieper, Christine Robson, and Amit Sheth. Context and Domain Knowledge Enhanced Entity Spotting in Informal Text. In Abraham Bernstein, David Karger, Tom Heath, Lee Feigenbaum, Diana Maynard, Enrico Motta, and Krishnaprasad Thirunarayan, editors, *The Semantic Web - ISWC 2009*, volume 5823 of *Lecture Notes in Computer Science*, pages 260–276. Springer Berlin / Heidelberg, 2009.

[GRG+11] Olivier Galibert, Sophie Rosset, Cyril Grouin, Pierre Zweigenbaum, and Ludovic Quintard. Structured and extended named entity evaluation in automatic speech transcriptions. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 518–526, Chiang Mai, Thailand, November 2011.

[GS96] Ralph Grishman and Beth Sundheim. Message Understanding Conference-6: a brief history. In *16th International Conference on Computational linguistics (COLING'96)*, pages 466–471, Copenhagen, Denmark, 1996.

[HAMA06] Joseph Hassell, Boanerges Aleman-Meza, and I. Arpinar. Ontology-Driven Automatic Entity Disambiguation in Unstructured Text. In Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 44–57. Springer Berlin / Heidelberg, 2006.

[HC06] Bin He and Kevin Chen-Chuan Chang. Automatic complex schema matching across Web query interfaces: A correlation mining approach. *ACM Trans. Database Syst.*, 31(1):346–395, March 2006.

[Hea92] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*, COLING '92, pages 539–545, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[HHD+07] Andreas Harth, Aidan Hogan, Renaud Delbru, Jürgen Umbrich, Stefan Decker, and Stefan Decker. SWSE: answers before links. 2007.

[HHMN99] Monika R. Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. Measuring index quality using random walks on the Web. *Computer Networks*, 31(11âĂŞ16):1291 – 1303, 1999.

[HHMN00] Monika R. Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. On near-uniform URL sampling. *Computer Networks*, 33(1-6):295 – 308, 2000.

[HHUD07a] Andreas Harth, Aidan Hogan, Jürgen Umbrich, and Stefan Decker. SWSE: Objects before documents!, 2007.

[HHUD07b] Aidan Hogan, Andreas Harth, Jürgen Umrich, and Stefan Decker. Towards a scalable search and query engine for the web. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 1301–1302, New York, NY, USA, 2007. ACM.

[HLA12] Sebastian Hellmann, Jens Lehmann, and Sören Auer. Towards an Ontology for Representing Strings. In *Proceedings of the EKAW 2012*, Lecture Notes in Artificial Intelligence (LNAI). Springer, 2012.

[HNN99] Allan Heydon, Marc Najork, and Marc Najork. Mercator: A Scalable, Extensible Web Crawler. pages 219–229, 1999.

[HPZC07] Bin He, Mitesh Patel, Zhen Zhang, and Kevin Chen-Chuan Chang. Accessing the deep web. *Commun. ACM*, 50(5):94–101, May 2007.

[HSC02] Siegfried Handschuh, Steffen Staab, and Fabio Ciravegna. S-CREAM ? Semi-automatic CREAtion of Metadata. In Asunción Gómez-Pérez and V. Benjamins, editors, *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, volume 2473 of *Lecture Notes in Computer Science*, pages 165–184. Springer Berlin / Heidelberg, 2002.

[HSM01] David J. Hand, Padhraic Smyth, and Heikki Mannila. *Principles of data mining*. MIT Press, Cambridge, MA, USA, 2001.

[HTTG10] Jian Huang, Pucktada Treeratpituk, Sarah M. Taylor, and C. Lee Giles. Enhancing cross document coreference of web documents with context similarity and very large scale text categorization. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 483–491, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[HUD06] Andreas Harth, Jürgen Umbrich, and Stefan Decker. MultiCrawler: A Pipelined Architecture for Crawling and Indexing Semantic Web Data. In Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 258–271. Springer Berlin / Heidelberg, 2006.

[HUHD07] Andreas Harth, Jürgen Umbrich, Aidan Hogan, and Stefan Decker. YARS2: A Federated Repository for Querying Graph Structured Data from the Web. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 211–224. Springer Berlin / Heidelberg, 2007.

[IHU+10] Robert Isele, Andreas Harth, Jurgen Umbrich, Christian Bizer, and Christian Bizer. LDspider: An open-source crawling framework for the Web of Linked Data. In *Poster at the International Semantic Web Conference (ISWC2010), Shanghai*, 2010.

[jav12] Java-Rdfa - RDFa parser. https://github.com/shellac/java-rdfa, 2012.

[JRW+12] Yushi Jing, Henry A. Rowley, Jingbin Wang, David Tsai, Chuck Rosenberg, and Michele Covell. Google image swirl: a large-scale content-based image visualization system. In *WWW (Companion Volume)'12*, pages 539–540, 2012.

[KC04] Rohit Khare and Doug Cutting. Nutch: A flexible and scalable open-source web search engine. Technical report, 2004.

[KCN+08] Tomas Kliegr, Krishna Chandramouli, Jan Nemrava, Vojtech Svatek, and Ebroul Izquierdo. Combining image captions and visual analysis for image concept classification. In *Proceedings of the 9th International Workshop on Multimedia Data Mining: held in conjunction with the ACM SIGKDD 2008*, MDM '08, pages 8–17, New York, NY, USA, 2008. ACM.

[KKPS02] J. Kahan, M.-R. Koivunen, E. Prud'Hommeaux, and R.R. Swick. Annotea: an open RDF infrastructure for shared Web annotations. *Computer Networks*, 39(5):589 – 608, 2002.

[KKR+11] Saurabh S. Kataria, Krishnan S. Kumar, Rajeev R. Rastogi, Prithviraj Sen, and Srinivasan H. Sengamedu. Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 1037–1045, New York, NY, USA, 2011. ACM.

[KL05] Sung Kim and Sang Lee. An Empirical Study on the Change of Web Pages. In Yanchun Zhang, Katsumi Tanaka, Jeffrey Yu, Shan Wang, and Minglu Li, editors, *Web Technologies Research and Development - APWeb 2005*, volume 3399 of *Lecture Notes in Computer Science*, pages 632–642. Springer Berlin / Heidelberg, 2005.

[Kli10] Tomáš Kliegr. Entity classification by bag of Wikipedia articles. In *Proceedings of the 3rd workshop on Ph.D. students in information and knowledge management*, PIKM '10, pages 67–74, New York, NY, USA, 2010. ACM.

[KLK06] Shin Kwon, Sang Lee, and Sung Kim. Effective Criteria for Web Page Changes. In Xiaofang Zhou, Jianzhong Li, Heng Shen, Masaru Kitsuregawa, and Yanchun Zhang, editors, *Frontiers of WWW Research and Development - APWeb 2006*, volume 3841 of *Lecture Notes in Computer Science*, pages 837–842. Springer Berlin / Heidelberg, 2006.

[Koe02] Wallace Koehler. Web page change and persistence-A four-year longitudinal study. *Journal of the American Society for Information Science and Technology*, 53(2):162–171, 2002.

[KPT+04] Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49 – 79, 2004.

[KR00] Adam Kilgarriff and Joseph Rosenzweig. Framework and Results for English SENSEVAL. *Special Issue on SENSEVAL. Computers and the Humanties*, pages 15–48, 2000.

[KSRC09] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 457–466, New York, NY, USA, 2009. ACM.

[Kus97a] Nicholas Kushmerick. *Wrapper induction for information extraction*. PhD thesis, 1997.

[Kus97b] Nicholas Kushmerick. Wrapper Induction for Information Extraction. In *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-97*, pages 729–737, 1997.

[LA08] Jing Li and Nigel M. Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10Ú12):1771 – 1787, 2008.

[Les86] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pages 24–26, New York, NY, USA, 1986. ACM.

[Ley02] Michael Ley. The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. In Alberto Laender and Arlindo Oliveira, editors, *String Processing and Information Retrieval*, volume 2476 of *Lecture Notes in Computer Science*, pages 481–486. Springer Berlin / Heidelberg, 2002.

[LGZ03] Bing Liu, Robert Grossman, and Yanhong Zhai. Mining data records in Web pages. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 601–606, New York, NY, USA, 2003. ACM.

[Liu07a] Bing Liu. Structured Data Extraction: Wrapper Generation. In M. J. Carey and S. Ceri, editors, *Web Data Mining*, Data-Centric Systems and Applications, pages 323–380. Springer Berlin Heidelberg, 2007.

[Liu07b] Bing Liu. *Web Data Mining Exploring Hyperlinks, Contents, and Usage Data*. Springer Berlin Heidelberg New York, 2007.

[LJM06] Hongyu Liu, Jeannette Janssen, and Evangelos Milios. Using HMM to learn user browsing patterns for focused Web crawling. *Data & Knowledge Engineering*, 59(2):270 – 291, 2006.

[LLM11] Berenike Litz, Hagen Langer, and Rainer Malaka. Sequential Supervised Learning for Hypernym Discovery from Wikipedia. In Ana Fred, Jan L. G. Dietz, Kecheng Liu, and Joaquim Filipe, editors, *Knowledge Discovery, Knowlege Engineering and Knowledge Management*, volume 128 of *Communications in Computer and Information Science*, pages 68–80. Springer-Verlag, Berlin Heidelberg, 2011.

[LLWL08] Hsin-Tsang Lee, Derek Leonard, Xiaoming Wang, and Dmitri Loguinov. IRLbot: scaling to 6 billion pages and beyond. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 427–436, New York, NY, USA, 2008. ACM.

[LMJ04] Hongyu Liu, Evangelos Milios, and Jeannette Janssen. Probabilistic models for focused web crawling. In *Proceedings of the 6th annual ACM international workshop on Web information and data management*, WIDM '04, pages 16–22, New York, NY, USA, 2004. ACM.

[LMP01] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[LPH00] Ling Liu, Calton Pu, and Wei Han. XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources. In *ICDE*, pages 611–621, 2000.

[LRNdS02] Alberto H.F. Laender, Berthier Ribeiro-Neto, and Altigran S. da Silva. DEByE - Data Extraction By Example. *Data & Knowledge Engineering*, 40(2):121 – 154, 2002.

[Mah36] P. C. Mahalanobis. On the generalised distance in statistics. *National Institute of Sciences of India*, 2(1):49âĂŞ55, 1936.

[MB00] Filippo Menczer and Richard K. Belew. Adaptive Retrieval Agents: Internalizing Local Context and Scaling up to the Web. *Machine Learning*, 39:203–242, 2000.

[MC07] 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*, CIKM '07, pages 233–242, New York, NY, USA, 2007. ACM.

[Mih05] Rada Mihalcea. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 411–418, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[Mil95] George A. Miller. WordNet: a lexical database for English. *Commun. ACM*, 38(11):39–41, November 1995.

[MJC+07] Jayant Madhavan, Shawn R. Jeffery, Shirley Cohen, Xin Dong, David Ko, Cong Yu, Alon Halevy, Google Inc, and Google Inc. Web-scale Data Integration: You Can Only Afford to Pay As You Go. In *In Proc. of CIDR-07*, 2007.

[MJGSB11a] 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*, I-Semantics '11, pages 1–8, New York, NY, USA, 2011. ACM.

[MJGSB11b] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *7th International Conference on Semantic Systems (I-Semantics)*, 2011.

[MKK+08] Jayant Madhavan, David Ko, Lucja Kot, Vignesh Ganapathy, Alex Rasmussen, and Alon Halevy. Google's Deep Web crawl. *Proc. VLDB Endow.*, 1(2):1241–1252, August 2008.

[MKS+04] G. Mohr, M. Kimpton, M. Stack, I. Ranitovic, and I. Ranitovic. Introduction to heritrix, an archival quality web crawler. In *4th International Web Archiving Workshop (IWAW04)*, 2004.

[ML03] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[MMK99] Ion Muslea, Steve Minton, and Craig Knoblock. A hierarchical approach to wrapper induction. In *Proceedings of the third annual conference on Autonomous Agents*, AGENTS '99, pages 190–197, New York, NY, USA, 1999. ACM.

[MPS04] Filippo Menczer, Gautam Pant, and Padmini Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. *ACM Trans. Internet Technol.*, 4(4):378–419, November 2004.

[MSA+10] Pekka Malo, Pyry Siitari, Oskar Ahlgren, Jyrki Wallenius, and Pekka Korhonen. Semantic Content Filtering with Wikipedia and Ontologies. In *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, ICDMW '10, pages 518–526, Washington, DC, USA, 2010. IEEE Computer Society.

[MW08a] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM.

[MW08b] David Milne and Ian H. Witten. Learning to link with Wikipedia. In *17th ACM International Conference on Information and Knowledge Management (CIKM'08)*, pages 509–518, Napa Valley, California, USA, 2008.

[MWM08] Olena Medelyan, Ian H. Witten, and David Milne. Topic Indexing with Wikipedia, 2008.

[NCO04] Alexandros Ntoulas, Junghoo Cho, and Christopher Olston. What's new on the web?: the evolution of the web from a search engine perspective. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 1–12, New York, NY, USA, 2004. ACM.

[NHNH01] Marc Najork, Allan Heydon, Marc Najork, and Allan Heydon. High-performance web crawling. Technical report, SRC Research Report 173, Compaq Systems Research, 2001.

[NL96] Hwee Tou Ng and Hian Beng Lee. Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, ACL '96, pages 40–47, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[NT89] Kumpati S. Narendra and Mandayam A. L. Thathachar. *Learning automata: an introduction*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[NTM06] David Nadeau, Peter Turney, and Stan Matwin. Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity. In Luc Lamontagne and Mario Marchand, editors, *Advances in Artificial Intelligence*, volume 4013 of *Lecture Notes in Computer Science*, pages 266–277. Springer Berlin / Heidelberg, 2006.

[Nto05] Alexandros Ntoulas. Downloading textual hidden web content through keyword queries. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 100–109, 2005.

[NV05] Roberto Navigli and Paola Velardi. Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1075–1086, July 2005.

[NW01] Marc Najork and Janet L. Wiener. Breadth-first crawling yields high-quality pages. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 114–118, New York, NY, USA, 2001. ACM.

[ODC+08] Eyal Oren, Renaud Delbru, Michele Catasta, Richard Cyganiak, and Giovanni Tummarello. Sindice.com: A document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies*, 3, 2008.

[Pas04] Marius Pasca. Acquisition of categorized named entities for web search. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 137–145, New York, NY, USA, 2004. ACM.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[PCB+09] Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*, EMNLP '09, pages 938–947, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[Ped01] Ted Pedersen. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–8, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics.

[PKK+03] Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, and Miroslav Goranov. KIM ? Semantic Annotation Platform. In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, *The Semantic Web - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 834–849. Springer Berlin / Heidelberg, 2003.

[PLB+06] Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, Alpa Jain, and Alpa Jain. Organizing and Searching the World Wide Web of Facts - Step One: The One-Million Fact Extraction Challenge. In *AAAI 2006*, 2006.

[PM02] Gautam Pant and Filippo Menczer. MySpiders: Evolve Your Own Intelligent Web Crawlers. *Autonomous Agents and Multi-Agent Systems*, 5:221–229, 2002.

[PP09] Marco Pennacchiotti and Patrick Pantel. Entity extraction via ensemble semantics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 238–247, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[PRA+02] Alberto Pan, Juan Raposo, Manuel Álvarez, Paula Montoto, Vicente Orjales, Justo Hidalgo, Lucía Ardao, Anastasio Molano, and Ángel Viña. The denodo data integration platform. In *Proceedings of the 28th international conference on Very Large Data Bases*, VLDB '02, pages 986–989. VLDB Endowment, 2002.

[PS05] Gautam Pant and Padmini Srinivasan. Learning to crawl: Comparing classification schemes. *ACM Trans. Inf. Syst.*, 23(4):430–462, October 2005.

[PS06] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/, 2006.

[PSMM02] Gautam Pant, Padmini Srinivasan, Filippo Menczer, and Filippo Menczer. Exploration versus Exploitation in Topic Driven Crawlers. 2002.

[QMTM04] Till Quack, Ullrich Mönich, Lars Thiele, and B. S. Manjunath. Cortina: a system for large-scale, content-based web image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 508–511, New York, NY, USA, 2004. ACM.

[RJ99] Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, AAAI '99/IAAI '99, pages 474–479, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.

[Row09] Matthew Rowe. Applying Semantic Social Graphs to Disambiguate Identity References. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 461–475. Springer Berlin / Heidelberg, 2009.

[RPA+02] Juan Raposo, Alberto Pan, Manuel Álvarez, Justo Hidalgo, and Ángel Viña. The Wargo System: Semi-Automatic Wrapper Generation in Presence of Complex Data Access Modes. In *Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, DEXA '02, pages 313–320, Washington, DC, USA, 2002. IEEE Computer Society.

[RSJ88] Stephen E. Robertson and Karen Sparck Jones. Document retrieval systems. chapter Relevance weighting of search terms, pages 143–160. Taylor Graham Publishing, London, UK, UK, 1988.

[RT11a] Giuseppe Rizzo and Raphaël Troncy. NERD: A Framework for Evaluating Named Entity Recognition Tools in the Web of Data. In *10th International Semantic Web Conference (ISWC'11), Demo Session*, pages 1–4, Bonn, Germany, 2011.

[RT11b] Giuseppe Rizzo and Raphaël Troncy. NERD: Evaluating Named Entity Recognition Tools in the Web of Data. In *Workshop on Web Scale Knowledge Extraction (WEKEX'11)*, pages 1–16, Bonn, Germany, 2011.

[RT12] Giuseppe Rizzo and Raphaël Troncy. NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools. In *13th Conference of the European Chapter of the Association for computational Linguistics (EACL'12)*, 2012.

[RTG98] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A Metric for Distributions with Applications to Image Databases. In *ICCV*, pages 59–66, 1998.

[RTHB12] Giuseppe Rizzo, Raphaël Troncy, Sebastian Hellmann, and Martin Bruemmer. NERD meets NIF: Lifting NLP Extraction Results to the Linked Data Cloud. In *5th International Workshop on Linked Data on the Web (LDOW'12)*, Lyon, France, 2012.

[SC96] John R. Smith and Shih-Fu Chang. VisualSEEk: a fully automated content-based image query system. In *Proceedings of the fourth ACM international conference on Multimedia*, MULTIMEDIA '96, pages 87–98, New York, NY, USA, 1996. ACM.

[SC12] Valentin I. Spitkovsky and Angel X. Chang. A cross-lingual dictionary for English Wikipedia concepts. In *Proceedings of LREC 2012*, 2012.

[SDB+07] Marta Sabou, Martin Dzbor, Claudio Baldassarre, Sofia Angeletou, and Enrico Motta. WATSON: A Gateway for the Semantic Web. In *Poster session of the European Semantic Web Conference, ESWC*, 2007.

[Sek98] Satoshi Sekine. NYU: Description of the Japanese NE system used for MET-2. In *Proceedings of Message Understanding Conference*, 1998.

[SJN05] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems*, number 17, pages 1297–1304, Cambridge, MA, 2005. MIT Press.

[SJN06] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 801–808, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[SMMP00] David McG. Squire, Wolfgang Müller, Henning Müller, and Thierry Pun. Content-based query of image databases: inspirations from text retrieval. *Pattern Recogn. Lett.*, 21(13-14):1193–1198, December 2000.

[SNN04] Satoshi Sekine, Chikashi Nobata, and Chikashi Nobata. Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In *LREC*, 2004.

[SS04] Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[Swa02] Aaron Swartz. Musicbrainz: A semantic web service. pages 76–77, 2002.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

[SZ03] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470 –1477 vol.2, oct. 2003.

[THS11] R. Troncy, B. Huet, and S. Schenk. *Multimedia Semantics: Metadata, Analysis and Interaction*. Wiley, 2011.

[TKSDM03] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[Tur01] Peter D. Turney. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, London, UK, UK, 2001. Springer-Verlag.

[VKMM07] Raphael Volz, Joachim Kleb, Wolfgang Mueller, and Wolfgang Mueller. Towards Ontology-based Disambiguation of Geographical Identifiers. In *I3*, 2007.

[WF99] Ian H. Witten and Eibe Frank. Data mining: practical machine learning tools and techniques with Java implementations. In *The Morgan Kaufmann series in data management systems*. Morgan Kaufmann, 1999.

[WKPU08] Casey Whitelaw, Alex Kehlenbeck, Nemanja Petrovic, and Lyle Ungar. Web-scale named entity recognition. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 123–132, New York, NY, USA, 2008. ACM.

[WWLM04] Jiying Wang, Ji-Rong Wen, Fred Lochovsky, and Wei-Ying Ma. Instance-based schema matching for web databases by domain-specific query probing. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 408–419. VLDB Endowment, 2004.

[wwwa] Apache Lucene. `http://lucene.apache.org/`. Accessed: 19/09/2012.

[wwwb] Apache Solr. `http://lucene.apache.org/solr/`. Accessed: 19/09/2012.

[wwwc] mnoGoSearch – Internet Search Engine Software. `http://www.mnogosearch.org/`. Accessed: 19/09/2012.

[wwwd] Sphinx | Open Source Search Server. `http://sphinxsearch.com/`. Accessed: 19/09/2012.

[WYDM04] Wensheng Wu, Clement Yu, AnHai Doan, and Weiyi Meng. An interactive clustering-based approach to integrating source query interfaces on the deep Web. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, SIGMOD '04, pages 95–106, New York, NY, USA, 2004. ACM.

[ZCNN05] Petros Zerfos, Junghoo Cho, Alexandros Ntoulas, and Alexandros Ntoulas. Downloading textual hidden web content through keyword queries. pages 100–109, 2005.

[ZIL12] Dengsheng Zhang, Md. Monirul Islam, and Guojun Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362, 2012.

[ZL05] Yanhong Zhai and Bing Liu. Extracting Web Data Using Instance-Based Learning. In Anne Ngu, Masaru Kitsuregawa, Erich Neuhold, Jen-Yao Chung, and Quan Sheng, editors, *Web Information Systems Engineering - WISE 2005*, volume 3806 of *Lecture Notes in Computer Science*, pages 318–331. Springer Berlin / Heidelberg, 2005.

[ZL06] Yanhong Zhai and Bing Liu. Structured Data Extraction from the Web Based on Partial Tree Alignment. *Knowledge and Data Engineering, IEEE Transactions on*, 18(12):1614 –1628, dec. 2006.

[ZMW+05] Hongkun Zhao, Weiyi Meng, Zonghuan Wu, Vijay Raghavan, and Clement Yu. Fully automatic wrapper generation for search engines. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 66–75, New York, NY, USA, 2005. ACM.

[ZNW+05] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. 2D Conditional Random Fields for Web information extraction. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 1044–1051, New York, NY, USA, 2005. ACM.