NADINE STEINMETZ

CONTEXT-AWARE SEMANTIC ANALYSIS OF VIDEO METADATA

CONTEXT-AWARE SEMANTIC ANALYSIS OF VIDEO METADATA

DISSERTATION
zur Erlangung des akademischen Grades
Doktor Rerum Naturalium (Dr. rer. nat.)
am Fachgebiet Internet-Technologien und -Systeme

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Universität Potsdam

von
NADINE STEINMETZ

Betreuer:	Prof. Christoph Meinel
Gutachter:	Prof. Sören Auer
	Prof. Steffen Staab

Datum der mündlichen Aussprache: 6. Mai 2014

Published online at the
Institutional Repository of the University of Potsdam:
URL http://opus.kobv.de/ubp/volltexte/2014/7055/
URN urn:nbn:de:kobv:517-opus-70551
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus-70551

Goethe sagte:
Leider lässt sich wahrhafte Dankbarkeit mit Worten nicht ausdrücken.
Und ich bin wahrhaft dankbar: meinem Ehemann, meiner Familie,
meinen Freunden.

ABSTRACT

The Semantic Web provides information contained in the World Wide Web as machine-readable facts. In comparison to a keyword-based inquiry, semantic search enables a more sophisticated exploration of web documents. By clarifying the meaning behind entities, search results are more precise and the semantics simultaneously enable an exploration of semantic relationships. However, unlike keyword searches, a semantic entity-focused search requires that web documents are annotated with semantic representations of common words and named entities. Manual semantic annotation of (web) documents is time-consuming; in response, automatic annotation services have emerged in recent years. These annotation services take continuous text as input, detect important key terms and named entities and annotate them with semantic entities contained in widely used semantic knowledge bases, such as Freebase or DBpedia.

Metadata of video documents require special attention. Semantic analysis approaches for continuous text cannot be applied, because a information of a context in video documents originates from multiple sources possessing different reliabilities and characteristics. This thesis presents a semantic analysis approach consisting of a context model and a disambiguation algorithm for video metadata. The context model takes into account the characteristics of video metadata and derives a confidence value for each metadata item. The lower the ambiguity and the higher the prospective correctness, the higher the confidence value. The metadata items derived from the video metadata are analyzed in a specific order from high to low confidence level. Previously analyzed metadata are used as reference points in the context for subsequent disambiguations. The contextually most relevant entity is identified by means of descriptive texts and semantic relationships to the context. The context is created dynamically for each metadata item, taking into account the confidence value and other characteristics. The proposed semantic analysis follows two hypotheses: metadata items of a context should be processed in descendent order of their confidence value, and the metadata that pertains to a context should be limited by content-based segmentation boundaries. The evaluation results support the proposed hypotheses and show increased recall and precision for annotated entities, especially for metadata that originates from sources with low reliability. The algorithms have been evaluated against several state-of-the-art annotation approaches. The presented semantic analysis process is integrated into a video analysis framework and has been successfully applied in several projects.

Im Vergleich zu einer stichwortbasierten Suche ermöglicht die semantische Suche ein präziseres und anspruchsvolleres Durchsuchen von (Web)-Dokumenten, weil durch die explizite Semantik Mehrdeutigkeiten von natürlicher Sprache vermieden und semantische Beziehungen in das Suchergebnis einbezogen werden können. Eine semantische, Entitäten-basierte Suche geht von einer Anfrage mit festgelegter Bedeutung aus und liefert nur Dokumente, die mit dieser Entität annotiert sind als Suchergebnis. Die wichtigste Voraussetzung für eine Entitäten-zentrierte Suche stellt die Annotation der Dokumente im Archiv mit Entitäten und Kategorien dar. Eine manuelle Annotation erfordert Domänenwissen und ist sehr zeitaufwendig. Die semantische Annotation von Videodokumenten erfordert besondere Aufmerksamkeit, da inhaltsbasierte Metadaten von Videos aus verschiedenen Quellen stammen, verschiedene Eigenschaften und Zuverlässigkeiten besitzen und daher nicht wie Fließtext behandelt werden können.

Die vorliegende Arbeit stellt einen semantischen Analyseprozess für Video-Metadaten vor. Die Eigenschaften der verschiedenen Metadatentypen werden analysiert und ein Konfidenzwert ermittelt. Dieser Wert spiegelt die Korrektheit und die wahrscheinliche Mehrdeutigkeit eines Metadatums wieder. Beginnend mit dem Metadatum mit dem höchsten Konfidenzwert wird der Analyseprozess innerhalb eines Kontexts in absteigender Reihenfolge des Konfidenzwerts durchgeführt. Die bereits analysierten Metadaten dienen als Referenzpunkt für die weiteren Analysen. So kann eine möglichst korrekte Analyse der heterogen strukturierten Daten eines Kontexts sichergestellt werden. Am Ende der Analyse eines Metadatums wird die für den Kontext relevanteste Entität aus einer Liste von Kandidaten identifiziert – das Metadatum wird disambiguiert. Der Kontext für die Disambiguierung wird für jedes Metadatum anhand der Eigenschaften und Konfidenzwerte zusammengestellt. Der vorgestellte Analyseprozess ist an zwei Hypothesen angelehnt: Um die Analyseergebnisse verbessern zu können, sollten die Metadaten eines Kontexts in absteigender Reihenfolge ihres Konfidenzwertes verarbeitet werden und die Kontextgrenzen von Videometadaten sollten durch Segmentgrenzen definiert werden, um möglichst Kontexte mit kohärentem Inhalt zu erhalten. Durch ausführliche Evaluationen konnten die gestellten Hypothesen bestätigt werden. Der Analyseprozess wurden gegen mehrere State-of-the-Art Methoden verglichen und erzielt verbesserte Ergebnisse in Bezug auf Recall und Precision, besonders für Metadaten, die aus weniger zuverlässigen Quellen stammen. Der Analyseprozess ist Teil eines Videoanalyse-Frameworks und wurde bereits erfolgreich in verschiedenen Projekten eingesetzt.

ACKNOWLEDGMENTS

I would like to express my appreciation and thanks to my advisor Prof. Dr. Christoph Meinel who always has an open door for problems – regarding research or personal issues. In particular, I am especially grateful to my supervisor Dr. Harald Sack. He always encouraged me to enhance my research and achieve high level results. His passion for movies and books contributed to demonstrate the interesting parts and fun facts of our research.

I also want to thank my colleagues who always contributed to intensive discussions about problems and research issues despite their own immense work load. A special thanks goes to Magnus for being the best office mate – in good and bad times.

It is often said that a main part of a dissertation is endurance. I would not have been able to endure if there weren't these great lunch times together with Michaela, Lutz, Matthias, Christian, Philipp, and Patrick (amongst others). They always listened and gave advice when I needed encouragement to endure.

The last weeks before the submission of a thesis are always stressful. Therefore, I am very grateful to Meike, Anna, and Philipp for reading my thesis, reviewing it and giving advice for the final touch.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI	Artificial Intelligence
ASR	Automatic Speech Recognition
CA	Co-Occurrence Analysis
CL	Computational Linguistics
CRF	Conditional Random Field
DL	Description Logics
HMM	Hidden Markov Model
IE	Information Extraction

LGA	Link Graph Analysis
NED	Named Entity Disambiguation
NER	Named Entity Recognition
NLP	Natural Language Processing
OCR	Optical Character Recognition
OWL	Web Ontology Language
POS	Part-of-Speech
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SPARQL	SPARQL Protocol and RDF Query Language
SW	Semantic Web
URI	Uniform Resource Identifier
WSD	Word Sense Disambiguation
WWW	World Wide Web

PREFIXES

dbp	http://dbpedia.org/resource/
dbo	http://dbpedia.org/ontology/
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#

1 INTRODUCTION

This chapter introduces the subject of the thesis, defines the motivation and research objectives, lists the contributions achieved by this work and explains the structure of the thesis.

1.1 PROBLEM DEFINITION

Semantic analysis of natural language text is an essential part of providing semantically annotated documents for a semantic search and classification of documents. The semantic annotation of documents requires more than a basic understanding of its content. A document's content must be analyzed and semantic relationships between linguistic components must be detected in order to arrive at the meaning of the text. During semantic annotations, a document can be labeled with distinct entities and categories. This thesis is focused on the analysis and annotation of video documents. For videos, the comprehension is supported by visual and auditive classifications, but the more essential and effective insight is provided by textual metadata. A significant disadvantage of the rich expressiveness of natural language information is its inherent ambiguity. Therefore, the process of semantic analysis comprises the detection, recognition and disambiguation of named entities and important key terms within textual information. Disambiguation denotes the process of determining the correct interpretation of an ambiguous term for a given context. Comprehension and correct interpretation of a document's content is only enabled by a correct semantic annotation.

In recent years, several annotation tools have emerged for the analysis of continuous text. These approaches can be applied to analyze documents, such as web pages or text documents. Meanwhile, videos are firmly on track to become "first class citizens" of the web [106], and the annotation of video documents entails additional and unique challenges as compared to the annotation of text documents. A content-based search of videos requires content-based metadata. The most reliable approach to annotated videos based on their content is to ask domain experts for transcripts, tags or classifications. Considering the large amount of videos created and published every year, this task might be regarded as a never-ending task if performed manually. Automatic extraction algorithms, such as Optical Character Recognition (OCR), Automatic Speech Recognition (ASR), or Visual Concept Detection (VCD) provide essential information about video content. As an added benefit, these algorithms mostly complete their

tasks in real-time. Yet, automatic sources of textual metadata cannot provide the same level of reliability as (authoritative) human sources. The accuracy rates for OCR and ASR algorithms range between 0.35 and 0.9 depending on the applied approach, quality and the type of video material. Therefore, one of the challenges of video annotation is the handling of different source reliabilities of metadata.

Context is a factor that is mandatory for general understanding of natural language. The context necessary for the disambiguation of ambiguous terms within a document is provided by all the surrounding information such as further metadata or textual content related to the same document or fragment under consideration. Depending on the context, information might take on different meanings and thus lead to different interpretations. Context influences the interpretation of information and should therefore be as accurate as possible. For videos, the contexts for the interpretation of the content-based metadata is heterogenous. Under the assumption that all metadata occurring within the same time slot of a video form a context, the information pertaining to the context is of different sources, types and reliabilities. Thereby, the context for interpretation of content-based video metadata is considered to be heterogenous and information pertaining to such a context should be handled with particular caution. Due to the low reliability of the automatic extraction algorithms, a context might contain incorrect information which can lead to a faulty interpretation when this context is applied.

Thus, the semantic analysis of video metadata deserves special attention. The next section summarizes the research objectives that have been derived from the described prerequisites and introduces the challenges faced during its development process.

1.2 RESEARCH OBJECTIVES AND CHALLENGES

The research objectives and challenges presented in this thesis are mainly focussed on the analysis of video documents. Most of the developed algorithms can be applied to other document types and integrated into various scenarios. The generalization of the presented approach is further discussed in Section 12.1. Under the assumption of the premises introduced in the previous section, semantic analysis of video metadata must deal with the following research objectives:

- Disambiguation of natural language text in general,
- Definition of an appropriate context to enable correct interpretation, and
- Processing of metadata with various characteristics and reliabilities.

During the development process of this thesis, these objectives have been researched and implemented successively.

The following research questions are addressed:

What metadata currently exists for videos and what is relevant for an content-related search? The research started with exploitation of video metadata and characterizing its different types. Most textual metadata is provided as continuous text or keywords, but user-generated tags require special attention and pre-processing. Following this assumption, a first requirement for the semantic analysis process is derived:

Is there a semantic analysis process that is able to handle different text types? State-of-the-art annotation tools and web services only process continuous text and do not distinguish different text types. Therefore, a new analysis method which includes pre-processing and special handling of different text types has been developed and is presented in this thesis. The proposed method is able to handle video metadata and to create a context independent from the type of textual information. Since the main challenge of a general semantic analysis is to interpret ambiguous information correctly, algorithms are required that make use of the context and exploit its potential. This leads to the next research question.

How can context be utilized? For the semantic analysis of natural language texts, a context consists of textual information that gives hints about the appropriate interpretation. The proposed disambiguation algorithm takes into account the semantic entities and entity candidates that can be derived from the context. Previously disambiguated information and the resulting assigned entity is used as a reference point for subsequent disambiguations. It is nonetheless possible that this information might originate from an unreliable source and might therefore be wrong for the given context. This entails subsequent incorrect disambiguation if this wrong information is taken into account as context. For the special application of semantic analysis of video metadata, the characteristics of different metadata must be taken into account:

Can a trust level be measured for the information provided as metadata? How is this confidence determined? For this purpose, a context model taking into account several characteristics of the metadata has been developed and is presented in this thesis. A confidence value is derived from these characteristics and used to bring information in a specific order starting with the one with the highest probable accuracy. The disambiguation of information is performed under consideration of the ordered items contained in the context.

Which information is taken into account for a disambiguation process to enable a correct interpretation? The disambiguation process calculates a score for every entity candidate which reflects the relevance of the entity candidate for the given context. This score, the confidence value derived from the context model, and other determined characteristics of the metadata enable the creation of a context dynamically. For every unit of information to be processed, this decision can be taken separately considering the relevant aspects of metadata.

These research questions and challenges have been solved by the proposed context model and semantic analysis process. The contributions are summarized in the next section.

1.3 CONTRIBUTIONS

This thesis contributes to the research field of automatic semantic annotation of video metadata, but its findings can also be applied to the analysis of text documents or other textual information originating from different sources and possessing different characteristics. The proposed context model and algorithms were developed, implemented and evaluated to show the scientific contributions of my approach. The implementation of my approach is part of a framework that has been applied in several projects. To summarize, the contributions of this thesis are the following:

- The development and implementation of a competitive algorithm for the semantic analysis of video metadata that can be applied to multiple knowledge bases.
- The algorithm has been developed for the video metadata, but can also be applied to simple continuous text.
- The development of a context model enables the evaluation of metadata from different sources and derivation of a confidence value representing prospective ambiguity and accuracy of the metadata information. The confidence value enables a ranking of metadata items. The disambiguation process is performed respecting this order starting with the most confident metadata. This approach enables the definition and application of negative context. Negative context is a novel concept that has not been previously applied for other disambiguation approaches.
- The context for the disambiguation process is created dynamically, taking into account the characteristics of the metadata and their confidence values.
- The overall approach has been evaluated against several state-of-the-art approaches and is able to improve analysis results,

especially on video metadata with prospective low reliability with respect to correctness.

In the course of the research for my thesis, the following articles have been composed and published[1]:

Reviewed Publications in Journals

- Joerg Waitelonis, Nadine Ludwig, Magnus Knuth and Harald Sack. Whoknows? - Evaluating Linked Data heuristics with a quiz that cleans up DBpedia. In *International Journal of Interactive Technology and Smart Education (ITSE), Volume 8 (Issue 3), 2011.*

Reviewed Publications in Conferences/Workshops

- Nadine Steinmetz, Magnus Knuth and Harald Sack. Statistical Analyses of Named Entity Disambiguation Benchmarks. In *Proceedings of 1st International Workshop on NLP & DBpedia at ISWC 2013, Sydney, Australia, 2013.* Best Rated Paper at the Workshop.

- Nadine Steinmetz and Harald Sack. About the Influence of Negative Context. In *Proceedings of 7th International Conference on Semantic Computing (ICSC 2013), Irvine, USA, 2013.*

- Nadine Steinmetz and Harald Sack. Semantic Multimedia Information Retrieval Based on Contextual Descriptions. In *Proceedings of the 10th Extended Semantic Web Conference (ESWC 2013), Montpellier, France, 2013.*

- Christian Hentschel, Harald Sack and Nadine Steinmetz. Cross-Dataset Learning of Visual Concepts. In *Proceedings of 10th Workshop on Adaptive Multimedia Retrieval: User, Context, and Feedback (AMR 2012), Copenhagen, Denmark, 2012.*

- Christian Hentschel, Johannes Hercher, Magnus Knuth, Johannes Osterhoff, Bernhard Quehl, Harald Sack, Nadine Steinmetz, Joerg Waitelonis and Haojin Yang. Open Up Cultural Heritage in Video Archives with Mediaglobe. In *Proceedings of 12th International Conference on Innovative Internet Community Services (I2CS 2012), Trondheim, Norway, 2012.*

- Nadine Ludwig and Harald Sack. Named Entity Recognition for User-Generated Tags. In *Proceedings of 8th International Workshop on Text-based Information Retrieval (TIR 2011), Toulouse, France, 2011.*

- Nadine Ludwig, Joerg Waitelonis, Magnus Knuth and Harald Sack. WhoKnows? - Evaluating Linked Data Heuristics with a

1 Please note that all articles written before 2012 have been published under my birthname *Nadine Ludwig*.

Quiz that Cleans Up DBpedia. In *Proceedings of 8th Extended Semantic Web Conference (ESWC 2011), Poster Session, Heraklion, Greece, 2011.*

- Magnus Knuth, Nadine Ludwig, Lina Wolf and Harald Sack. The Generation of User Interest Profiles from Semantic Quiz Games. In *Proceedings of The Second International Workshop on Mining Ubiquitous and Social Environments (MUSE 2011), Athens, Greece, 2011.*

- Joerg Waitelonis, Nadine Ludwig and Harald Sack. Use What You Have – Yovisto Video Search Engine Takes a Semantic Turn. In *Proceedings of 5th International Conference on Semantic and Digital Media (SAMT), Saarbrücken, Germany, 2010.*

1.4 THESIS STRUCTURE

This thesis consists of three parts:

1. Foundations and Related Work
2. Semantic Analysis of Video Metadata
3. Evaluation and Conclusion

FOUNDATIONS AND RELATED WORK The first part of the thesis introduces technological and theoretical foundations as well as related work for the proposed approach. The thesis deals with natural language processing of video metadata and has been developed within the scope of the Semantic Web. Context is an important topic as it pertains to the thesis and defining context constitutes a major part of the work. Therefore, the first part of the thesis consists of these four chapters: *Video Metadata*, *Natural Language Processing*, *Semantic Web Technologies* and *Definitions of Context*.

- Chapter 2 deals with the general characteristics of video documents, and metadata available for the description of its content.

- Chapter 3 introduces the research field of Natural Language Processing and describes several approaches that have been developed to enable computers to understand natural language. Some of the approaches described in this chapter have been applied in the present thesis or further developed for the purpose of semantic analysis on video metadata within the scope of the Semantic Web.

- The main goals of the Semantic Web initiative as well as its technologies are introduced in Chapter 4. Section 4.3 introduces related work on semantic annotation of textual information. The

approach presented in this thesis has been evaluated against state-of-the-art tools that are introduced in this section.

- Chapters 5 deals with several definitions of context as developed in other research fields. This chapter gives an introduction to manyfold ways of interpreting and defining context.

SEMANTIC ANALYSIS OF VIDEO METADATA The second part of the thesis constitutes the main emphasis and introduces the developed context model as well as the disambiguation algorithm for the purpose of semantic analysis of video metadata.

- Chapter 6 introduces the characteristics and definition of context as applied for this thesis. For semantic analysis, the application of context characteristics as boundaries and manual or automatic refinement influences the quality of the result. Also, the chapter discusses positive and negative context as novel definitions. The previously mentioned *heterogenous context* is introduced and defined in Section 6.5.

- The developed context model is introduced in Chapter 7. The context model has been designed to handle different characteristics of video metadata. It takes into account several aspects of metadata and deduces a confidence value that is applied in the subsequent analysis process.

- Chapter 8 introduces the semantic analysis process and the disambiguation algorithm. The process handles textual information and assigns specific semantic entities to named entities and important key terms. The algorithm requires a semantic knowledge base to distinguish semantic entities and decide which entities are relevant for a given context. The knowledge base and its creation is described in Section 8.1. The disambiguation algorithm consists of several separate analyses taking into account a given context. The overall process and its constituents are outlined in Section 8.2.

- The application of the context model for the semantic analysis is introduced in Chapter 9. The confidence value derived from the context model is used to create a context for the disambiguation process dynamically and orders the items pertaining to the context according to their confidence values. Thereby, a disambiguation process beginning with the most confident item is achieved. This approach enables the creation of a negative context which is described in Section 9.2.2. The chapter concludes with an overview of the entire semantic analysis process and complexity examinations.

EVALUATION & APPLICATIONS In the third part of the thesis, the semantic analysis process is evaluated with respect to the quality of the achieved results compared to simple approaches and state-of-the-art tools for continuous texts. The proposed approach has been integrated into the processing chain of videos in several projects. The applications are introduced in Section 11.

- Initially, Chapter 10 outlines the evaluation objectives and applied measures. Several benchmarks and dictionaries for entity look-up are introduced and analyzed for multiple characteristics. Section 10.7 details evaluations for the separate approaches presented in this thesis. The semantic analysis process under consideration of the context model has been evaluated against several state-of-the-art approaches; a hypothesis test and significance analysis have been performed to prove the high quality of the results. The chapter is concluded by a discussion of the evaluation results.
- Chapter 11 introduces the projects which have applied the presented analysis process.

Chapter 12 concludes the thesis with a summary of the approach and a discussion on future work.

Part I

FOUNDATIONS AND RELATED WORK

2 VIDEO METADATA

This work is focused on the semantic analysis of video metadata. The following sections explain the demand of metadata in the scope of content-based semantic annotation of videos. Different types and characteristics of metadata are introduced and determined. Various characteristics of user-generated tags provided as video metadata are described. Additionally, several automatic extraction algorithms which provide useful information about a video's content are introduced.

2.1 VIDEO AND ITS CHARACTERISTICS REGARDING WEB SEARCH

Since the end of the 20th century, video portals have enriched the World Wide Web (WWW) by providing interesting information and entertainment via videos. The first video hosting site was *shareyourworld.com*[1]. It was founded in 1997, but was closed by 2001. Presently, the most popular internet video hosting site is *Youtube*[2]. In 2007, eight hours of video material were uploaded per minute, whereas in 2013 the upload rate amounted to 100 hours[3]. These numbers show the increasing interest in videos on the web, but also the increasing amount of valuable information available in terms of videos. The video portals are challenged to make the videos accessible and searchable, which is where the specific characteristics of video comes into play. Often, only minimal textual information is provided by the author or uploader of a video. Useful information describing the video content is rare, and content-related information in terms of information about what happens or is mentioned at which position in the video is most often completely missing. Therefore, a content-related search of videos is difficult. Video metadata describing the video and its content are nonetheless essential for a video search on the web. The next section introduces the term metadata and classifies different types.

2.2 DEFINITION AND CLASSIFICATION OF METADATA

Metadata can be defined as *data about data*. However, ISO/IEC 11379 states that this phrase is inaccurate and incomplete, because metadata can be descriptive data about things other than data, and all data is about some other data [38]. Thus, metadata is data that describe in-

1 http://www.beet.tv/2007/07/first-video-sha.html
2 http://www.youtube.com
3 http://www.youtube.com/yt/press/en/statistics.html

formation resources, such as data or objects. Furthermore, it is stated that:

> [...] metadata is defined to be data that defines and describes other data. This means that metadata are data, and data become metadata when they are used in this way. This happens under particular circumstances, for particular purposes, and with certain perspectives, as no data are always metadata. The set of circumstances, purposes, or perspectives for which some data are used as metadata is called the context. So, metadata are data about data in some context. [39]

In general, metadata are classified as either *administrative*, *structural* or *descriptive* metadata [86].

ADMINISTRATIVE METADATA refer to information required for the management of the resource, such as creation data and place, file type, rights and preservation information.

STRUCTURAL METADATA provide information about how the components of the resource are organized and ordered.

DESCRIPTIVE METADATA refer to information about the content of the resource for purposes such as search and identification.

For multimedia documents, these three metadata types are further differentiated [96]. Table 1 shows the types and subtypes of metadata for video documents. Descriptive metadata fall into three categories: context-related, context-descriptive and object-descriptive (textual and non-textual) metadata. Structural metadata are classified as either feature or segment specification metadata. Administrative metadata are further differentiated into presentation-related, recording-related and storage-related metadata.

For content-related searches of video documents (see Section 2.1) content-descriptive metadata are required. Thus, object-descriptive textual metadata is applied to enable a content-related search. Annotations can either be added manually or extracted automatically. Manually appended annotations are referred to as user-generated tags. Definitions and characteristics of tags are presented in Section 2.3.

Automatic extraction of content-based textual information can be performed by either detecting displayed text (via Optical Character Recognition (OCR)) or by eliminating the audio stream and extracting spoken language (via Automatic Speech Recognition (ASR)). The approaches of OCR and ASR are briefly introduced in Sections 2.4.2 and 2.4.3.

Table 1: Classification of metadata for video documents [96].

Class		Example
content-descriptive (interpreting)	object-descriptive (non-textual)	persons, objects, activities, title, impressions
	object-descriptive (textual)	annotation, script, subtitles
	context-related	identification, spatial and time information
	context-descriptive	vocabulary, thesauri, ontologies
structural (content-related – not interpreting)	feature	color distribution, texture, sound dynamics, form
	segment specification	start and end of a scene, bounding box of a frame fragment
administrative	presentation-related	QoS, resolution, layout
	recording-related	author, recording system
	storage-related	media type, storage format, storage location

Structural information about the video enables content-based pointing at textual information. Thereby, an annotation is assigned to a timestamp within the video and the timestamp belongs to structural subdivision. The subdivision constitutes a fragment of the video of coherent content. The subdivision represents the context of the annotation. The reference to an annotation should include the context in which the annotation has been made. Therefore, structural metadata enable context-related analysis of the textual metadata and also context-related search within the video. The extraction of structural information with respect to video documents is briefly introduced in Section 2.4.1.

In the next section, content-based metadata in terms of tags provided by users of a video portal are introduced.

2.3 USER-GENERATED TAGS

Social bookmarking services such as Delicious[4] derive their success from various community functionalities, first and foremost among them the ability of users to tag their own and other resources. In this way, a huge amount of valuable user-generated metadata is created. This metadata is essential to enable an efficient search within a portal.

User-generated tags can be characterized as keywords, category names, or metadata. Any user of a portal can tag any resource within

4 http://www.delicious.com

the limit of user-specified permissions [94]. Users don't follow any formal guidelines which results in variation in how resources are tagged. Resources can be tagged with any term that, from the user's point of view, represents a relationship between the resource and a concept [113]. Within three identified categories of intended audiences of the tags (Self, Family & Friends, Public), the category Public is ranked as the most important motivation for tagging [85]. In other words, a user's intention behind tagging is most likely motivated by providing descriptive information to make the resource trackable. Furthermore, Golder et al. identified seven different tag functions [40]:

- Identifying what (or who) it is about
- Identifying what it is
- Identifying who owns it
- Refining categories
- Identifying qualities and characteristics
- Self reference
- Task organizing

Most of these functions imply that tags can describe a resource on different levels of abstraction. Tags can explicitly name an entity, or can be descriptive regarding the category to which the tagged resource belongs. In this way, a semantic search (see Section 4.2) both on an entity and a category level can be enabled by semantically enriched user-generated tags.

According to a study about structure and characteristics of folksonomy tags, an average of 83% of user-generated tags are single terms [102]. Furthermore, an average of 82% of the reviewed tags are nouns. Based on these results, tag processing for composite terms, such as camel case ("barackObama") might be ignored. Single tags might be considered as subjects or categories describing a resource. As a tag may also be part of a group of nouns representing a single entity ("flying machine","albert einstein"), the tags stored as single words without any given order must be combined into tag groups of two or more terms to enable a mapping to all appropriate entities. Hence, each simple tag or group of tags within a given context may represent a distinct entity. A tag combination process and subsequent mapping of terms and term groups to entities are described in Section 8.2.1. The number of terms to be combined influences the complexity of the process. For this purpose, statistics on token counts of labels of entities have been calculated. Therefore, four dictionaries have been evaluated for the number of tokens of the containing labels. Results are presented in Section 10.6.4.

2.4 INFORMATION EXTRACTION FROM VIDEOS

The purpose of information extraction is to analyze multimedia, in this case videos, to excerpt content for a particular scope [69].

The information extraction approaches for video documents relevant for the presented work are:

- structural segmentation
- Optical Character Recognition
- Automated Speech Recognition
- detection of visual concepts

Structural segmentation divides a video document into coherent parts with respect to the content. The underlying assumption is that scene cuts mark a transition between two different contexts. One part of the video ends and a new part containing information about a different subject and representing a different context begins. The videos are segmented according to visual characteristics. A segmentation can also be achieved by analyzing the audio stream of the video. Pauses might mark the transition between two segments. This type of video segmentation is dependent upon the availability of audio information, but not all videos are provided with an audio stream, such as educational videos or screen casts. Section 2.4.1 gives a brief introduction to structural segmentation approaches based on visual features.

OCR extracts textual information from a video. The textual information can either be overlay text, such as subtitles, or in scene text, such as the text of advertisement boards displayed in the videos. This research field is briefly introduced in Section 2.4.2.

ASR converts spoken language to text and aligns it to the video document. Similar to OCR text, ASR texts provide essential information about the video content. The spoken text might describe in more detail what can be seen visually, but it might also differ from the visual content. Nevertheless, visual and spoken information that occur together in a video are considered to be correlated. Typically, ASR text is more precise and gives more context information than OCR text. An introduction to ASR is given in Section 2.4.3.

The detection of visual concepts provides more information about what can actually be seen in the video. Therefore, visual concepts can provide important context information supporting other analysis processes. A visual concept classifies an image or video segment and might constitute several functions: describe the scene (daytime, night, city scape, etc.), depict objects (airplanes, cars, ships, persons, etc.), reflect the content (graffiti, lecture slides, auditorium, etc.), or identify quality issues (overexposed, blurry, etc.)[5]. For instance, the

5 http://www.imageclef.org/2011/photo

visual concept *person portrait* is extracted. At the same time, OCR extracts overlay text at the bottom of the video frame and the voice-over mentions a person. Therefore, it might be presumed that the visual concept, the OCR text and the voice-over refer to the same person and altogether provide important contextual information. Section 2.4.4 introduces recent approaches to the detection of visual concepts and describes useful concepts for the purpose of this work.

2.4.1 *Structural Segmentation*

For the purpose of video content analysis, the detection of scene cuts constitutes the initial step. Typically, a video is shot in a sequence of successive frames originating from a single recording. The detection of shot boundaries provides useful information about the temporal structure of a video and constitutes the basis for the extraction of representative keyframes or frame candidates for video OCR and visual concept detection [48]. Two different types of shot boundary transitions can be distinguished: abrupt scene changes (hard cuts) and gradual transitions extending to more than one frame (soft cuts) [5].

Soft cuts can be further sub-divided into *fade-ins*, *fade-outs*, *wipes*, and *dissolves*.

Often, only hard cuts and fades have been considered for detection due to the rare appearance of the other types of cuts in the considered videos [48]. For hard cuts, Adjeroh et al. introduced an approach based on pixel differences of consecutive frames [2]. The statistical method computes the L2-norm between every five consecutive frames. A hard-cut candidate is considered if the gradient computed on the first derivative of this metric exceeds a specific threshold. The threshold has been determined empirically [48]. In contrast, faded scene cuts are harder to detect, because they feature a successive change in illumination, typically ending or starting with a black frame. For the purpose of detecting fades in videos, the algorithm is based on the entropy of the image. A fade-in is characterized by a monotonously increasing entropy. A local minimum and maximum of the entropy determines the beginning and end frame for the considered fade respectively.

2.4.2 *Optical Character Recognition*

Standard OCR approaches typically strive to extract textual data from scans of printed documents. These approaches need to be extended in order to extract textual data from video documents. A recent approach for video OCR has been developed in the course of the projects introduced in Section 11 and consists of two steps [48]:

- text detection: keyframe identification containing text, and

- text recognition: determination and recognition of text within the identified frames.

The first step distinguishes between video frames that prospectively contain text and those that do not contain text. This distinction is rather essential, because a video consists of a high of number of frames and an OCR process on all video frames might entail a time-consuming algorithm and noisy results. The subsequent step aims to separate pixels that belong to text from background pixels. The results of the latter step are provided by a standard OCR engine. The classification of text candidates is performed by applying a fast edge-based multi-scale detector. Subsequently, a projection-profiling algorithm is applied on each keyframe and regions possessing low edge density are identified as background regions and rejected. For further refinement of text regions in a frame, an adapted Stroke Width Transform (SWT) algorithm is applied [123]. To overcome the problem of distinguishing between heterogeneous background with low contrast and actual text pixels, appropriate binarization techniques have been applied. Yang et al. introduced a novel skeleton-based approach for video text binarization [124]. As a result, the text region is converted to black text pixels on a white background and provided to the standard OCR software (*Tesseract*[6]) for text recognition. Traditional spell checking algorithms are based on the assumption that spelling errors originate from typing errors. Thus, a spell checker for OCR must be based on the visual similarities of characters rather than the closeness of two keys on a keyboard or typewriter. Therefore, an adapted version of the open source *Hunspell*[7] spell checking algorithm must be applied to improve the quality of the achieved OCR results.

The described method achieves an F_1-measure of up to 93% on an independent test set [123], bearing in mind that these results are achieved on a dataset of extracted frames. Results for video OCR taking into account all frames of a video are typically much lower with a precision as low as 35 % [119]. This leads to the conclusion that video metadata originating from OCR sources possess a low reliability with respect to accuracy. This issue is considered in the presented context model and further discussed in Section 7.1.

2.4.3 *Automatic Speech Recognition*

Automatic Speech Recognition (ASR) enables machines to identity components of human speech and convert them into text. The process mainly is twofold: feature extraction and classification. The extracted features of an audio stream have to meet several requirements in order for speech to be identified and segmented. Martens et al. defined three main requirements [68]:

6 http://code.google.com/p/tesseract-ocr/
7 http://hunspell.sourceforge.net/

- allow the discrimination of similar sounding speech sounds
- allow models to be created without the need for an exorbitant amount of training data
- suppress specific characteristics of speaker and environment

The most common features extracted from the audio stream are Mel-Frequency Cepstral Coefficients (MFCC). MFCC are based on a linear cosine transformation of a log-spaced power spectrum on a nonlinear mel-scale of the frequency [29].

The sound signal is transformed into feature vectors and subsequently the vectors are classified and the speech is recognized. For the classification, several approaches have been developed:

- Pattern matching
- Neural networks
- Knowledge-based approaches
- Hidden Markov Model (HMM)

Most recent ASR algorithms are based on HMM, because speech can be considered as a Markov model for several stochastic purposes [29].

Word error rates of recent ASR approaches range between 10% and 50% [24]. These results influence the reliability of text originating from ASR sources and are further discussed in Section 7.1 wherein the context model is introduced and the source reliability of video metadata is considered.

2.4.4 *Detection of Visual Concepts*

Besides ASR and OCR, the detection of depicted visual concepts provides additional information about the video content and the resulting context.

Similar to the video segmentation and OCR, several approaches for visual concept detection have been developed in the scope of the projects introduced in Section 11. The most recent approach employs methods for content-based image classification to classify key frames of video segments into visual concept classes. The approach follows the *bag of keypoints* model [21], where the low-level visual features of a keyframe are described by local image patterns. For this, Scale Invariant Feature Transform (SIFT) features are extracted on each channel of a keyframe and used to generate a set of representative visual codewords by running a k-means clustering algorithm [48].

For the classification of a key frame, a kernel-based Support Vector Machine (SVM) is applied. The SVM classifier has been trained using labeled data. For each visual concept to be classified, a separate SVM is trained. Hence, the classifier is trained to solve a binary

classification problem, such as whether or not a keyframe depicts the considered visual concept. Taking into account the bag of keypoints feature vector and a SVM model trained for a specific visual concept keyframes can be classified into every possible visual concept class. For instance, the following visual concept classes might be useful for the search in video archives:

- day/night scene
- indoor/outdoor
- person/group of persons
- lecture/auditorium

The approach on visual concept detection described above can be extended to additional concepts simply by training additional SVM classifiers.

As already mentioned, in some cases visual concepts can provide important context information which help to interpret the textual metadata occurring at the same time in the video. These cases are very specific and need particular attention, however visual concepts as part of heterogenous contexts (see Section 6.5) in video metadata are not part of the presented context model, but their applications are briefly discussed in Section 12.2 (Future Work).

3 NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP)[1] refers to a research field focused on the understanding of the structure and meaning of natural language by computers. The challenge is to *decode* natural language to make it accessible for machine readability. For this purpose, several analysis steps have been developed. They are aggregated as text segmentation, syntactic analysis, and semantic analysis. Most methods aim at the detection, identification, and/or classification of named entities. This chapter introduces and describes these analyses and their respective subtasks.

3.1 TEXT SEGMENTATION

Text segmentation involves the decoding of the structure of natural language and therefore the division of text into linguistically meaningful parts [88]. Continuous text might be divided into smaller units – and thereby smaller contexts – by the detection of separate sentences. For the detection of named entities, the sentences need to be tokenized and word boundaries determined.

Sentence detection is also referred to as sentence boundary detection/disambiguation/recognition, in which continuous text is divided into individual sentences [88]. For western writing systems, a period typically marks the end of a sentence, but they can also be used for abbreviations and in numbers. Therefore, sentence boundary detection and word boundary detection cannot be considered independent tasks. Palmer lists four dependencies when developing text segmentation algorithms: language dependence, character-set dependence, application dependence and corpus dependence [88]. Obviously, different languages use different punctuation marks or symbols to denote the end of a sentence. The determination of the character-set is essential to identify single characters and assign them to the known characters of the underlying writing system. Text segmentation is mainly performed using trained models; any model is dependent on the corpora used during training. For instance, the English possessive *'s* is treated differently in different annotated corpora. Sometimes it is annotated separately from the preceding noun and sometimes the noun and the possessive are annotated together as

1 NLP is also sometimes referred to as Computational Linguistics (CL). Other sources claim a distinction between the two research fields is the respective intention: NLP is considered a type of engineering while CL is a branch of linguistics (science).

a possessive noun [41]. Furthermore, one must distinguish between written and spoken language.

3.2 SYNTACTIC AND SEMANTIC ANALYSIS

Text segmentation is only the first step in processing natural language, but essential for subsequent analysis steps. Comprehension of natural language is considered the ultimate goal for humans as well as NLP systems [90]. For the successful automatic interpretation, several steps are required: identification, classification and disambiguation of named entities. The necessary methods are described in the following sections.

3.2.1 *Part-of-Speech Tagging*

The identification of named entities within a continuous text is a first step in order to comprehend its content. The term *named entity* was first coined in the research field of Information Extraction (IE) in the 1990s [98]. In the course of the extraction of structured information from unstructured text, the recognition of names of persons, organizations, etc. is important. Therefore, the determination of the part of speech of the words is an important precursor [17]. Dependent on the particular purpose, named entities might be composed from words of different parts of speech, such as only nouns or nouns combined with adjectives. Part-of-speech tagging identifies the types of words in continuous text, sufficient for the subsequent process to extract the words of interest and prospective named entities.

The two main challenges in Part-of-Speech (POS) tagging are

- ambiguities of words with respect to their POS
- new and made-up words

Regarding ambiguity, the word *can* might be a noun, a verb or an auxiliary verb. The correct assignment has to be determined from the context. A more difficult challenge arises with new and made-up words. The Oxford Dictionary publishes quarterly overviews of newly added words[2]. For example, in 2013 the words *buzzworthy*, *cake pop*, *fauxhawk* and many others were added to the online dictionary. Again, the determination of the POS of new words can be done based on contextual information, as well as from information about the word itself, such as affixes [17].

For instance, the POS tagging library provided by the Stanford NLP Group[3] gives the following tagged result for the sentence *Steve McQueen drove down Sunset Boulevard in his Jaguar car.*:

2 http://blog.oxforddictionaries.com/

3 http://nlp.stanford.edu/software/tagger.shtml

Steve/NNP McQueen/NNP drove/VBD down/RP
Sunset/NNP Boulevard/NNP in/IN his/PRP
Jaguar/NNP car./NN
For this example, the annotated tags stand for:

- NNP – proper noun, singular
- VBD – verb, paste tense
- RP – particle
- IN – preposition or subordinating conjunction
- PRP – personal pronoun
- NN – noun, singular or mass

The provided tags depend on the annotated corpus utilized for the training. The Stanford POS tag vocabulary is based on the Penn Treebank tag set. The full list of tags contained in the tag set is provided by Marcus et al. [67].

One of the first attempts at automatic tagging of parts-of-speech was performed in 1963 by Klein and Simmons [55]. Their approach was based on a lexicon to look up prospective POS candidates for a word and a set of manually created rules. All classes found in the lexicon for the considered word were assigned as candidates. Subsequently, the rules were applied to remove POS candidates. In best case, only one candidate remains as the most probable POS of the word.

With the emergence of large annotated text corpora, the development of automatically trained POS taggers has become possible. The two most commonly used text corpora for American English are the Brown Corpus [33] and the Penn Treebank Corpus [67]. Based on these corpora, several statistical approaches have been developed and published. For example, Church introduced an approach based on Markov models in 1988 [19]. Similar approaches achieved up to 97% accuracy in English POS tagging with a training set containing about 10^6 words [17].

For many languages, such corpora as mentioned above are not available. The Baum-Welch algorithm enables the training of a Markov model without requiring a manually annotated corpus [7]. This approach enables an unsupervised tagging of texts. The algorithm first assigns initial probabilities to all parameters. Subsequently, the probabilities of the parameters are adjusted to increase the probability the model assigns to the training set until the training converges.

Other and more recent approaches are based on classical machine learning algorithms, such as transformation-based tagging [16], and maximum entropy conditional sequence models [115]. The latter approach combines the idea of maximum entropy with bidirectional dependency networks to take into account both preceding and subsequent contexts of a word to determine the POS. This approach has

been implemented as a Java library and provided by the Stanford NLP Group as *Stanford POS tagger*. This library is applied in the presented work to identify words' POS and subsequently detect prospective named entities and important terms.

3.2.2 *Named Entity Recognition*

Named Entity Recognition (NER) is also referred to as Named Entity Classification and aims to assign predefined classes, such as *person*, *location*, *organization* or *date* to named entities detected in continuous text. For instance, the sentence *Steve McQueen drove through Los Angeles in his Jaguar.* is annotated by the Stanford NER tagger[4] as follows:

<PERSON>Steve McQueen</PERSON> drove through
<LOCATION>Los Angeles</LOCATION> in his Jaguar.

NER approaches are mainly based on statistical models using the local structure of a text to make a decision on the classification. Published statistical sequence models include Hidden Markov Models [59], Conditional Markov Models [15], and Conditional Random Field (CRF) [56]. These approaches only take into account local characteristics for the classification decision. In 2005, Finkel et al. presented an approach to also incorporate non-local information into information extraction tasks [31]. The approach is based on an existing CRF system augmented with long-distance dependency models. Thereby, the consistency of already labelled entities occurring again later in the text is pursued. The presented work utilizes the library based on this approach (see Section 8.2.1 for more details on the use of the NER tagger).

3.2.3 *Word Sense Disambiguation*

Word Sense Disambiguation (WSD) aims to identify the correct meaning of a word in the given context. The main challenge for this task is the high level of ambiguity of natural language. WSD utilizes the results of POS tagging and named entity classification. The POS tagger identifies the word types within a sentence and subsequently the detected named entities – composed of different word types – are classified. Those classified named entities might still be ambiguous within the assigned class. For example, there are several persons with the name *Michael Jackson*: the famous pop singer, a journalist, a basketball player, etc. WSD attempts to remove the ambiguity and identify the most relevant entity for the given context. Furthermore, besides the disambiguation of named entities within a text, WSD also aims

4 http://nlp.stanford.edu/software/CRF-NER.shtml

to disambiguate common words. For instance, consider the following example sentences:

- I would like to have the grilled bass.
- Bill likes to play the bass.

The contexts of the two sentences allow different interpretations of the word *bass*: a fish or an instrument. Most of the time, humans do not have problems interpreting ambiguous information if a little context is given. Automatic disambiguation of unstructured textual information by machines, however, has been identified as an AI-complete problem [65]. Thus, the challenge is equivalent to essential problems of Artificial Intelligence (AI) and cannot be solved by a simple specific algorithm. The difficulty of the task originates from several factors rather than from a single cause [82]. Beside other factors, Navigli lists the dependency of knowledge of the WSD process as the main challenge. The process requires knowledge to assign relevant senses to the words of a text. For the actual disambiguation process, knowledge about the assigned senses is essential to distinguish them with respect to the given context; the latter type of knowledge can also be considered experience. Therefore, knowledge and experience are fundamental for a successful WSD process[5]. In reality, the creation and maintenance of a knowledge base is expensive and time-consuming and has been described as the *knowledge acquisition bottleneck* [35]. The task of WSD consists of four steps [82]:

- selection of word senses,
- application of external knowledge sources,
- representation of context, and
- automatic classification.

The basis of word sense disambiguation is a dictionary containing senses for all possible words. The main challenge is to distinguish between different senses and define the representation of the senses. In general, there are two different types of dictionaries: structured and unstructured resources. Structured resources include thesauri, machine-readable dictionaries and ontologies. Unstructured resources include raw and sense-annotated corpora, and collocation resources [82]. A widely used representative of machine-readable dictionaries is WordNet[6]. It can also be considered as a computational lexicon of the English language. WordNet represents different senses of words as different sets of synonyms – so-called *synsets* – describing the different senses separated by different word types, such as noun, verb, adjective, etc. For instance, the concept *jaguar* is represented by the following synset:

5 This assumption applies for both humans and computers.
6 `http://wordnet.princeton.edu`

Table 2: Different context sizes for the word *jaguar*

Context Size	Example
left neighbor (bigram)	... latest *Jaguar* ...
right neighbor (bigram)	... *Jaguar* cars ...
trigram	... while *jaguars* live ... ... *Jaguar* classic vehicles ... ... the black *jaguar* ...
sentence	... The *jaguar* is the third-largest feline after the tiger and the lion. ...
paragraph	... The *jaguar* is a near threatened species and its numbers are declining. Threats include loss and fragmentation of habitat. While international trade in *jaguars* or their parts is prohibited, the cat is still frequently killed by humans, particularly in conflicts with ranchers and farmers in South America. Although reduced, its range remains large; given its historical distribution, the *jaguar* has featured prominently in the mythology of numerous indigenous American cultures, including those of the Maya and Aztec.[7]...

```
{jaguar, panther, Panthera onca, Felis onca}
```

For every synset, a small descriptive text is provided, sometimes supplemented by an example for the use of the concept in a sentence. Thereby, the dictionary provides external knowledge about the word senses. The context for a WSD process on a word can be derived from preprocessing steps, such as POS tagging, and NER, as well as from the information surrounding the word. Therefore, different context sizes might be considered. For continuous text, the context size ranges from left/right neighbor, over n-grams, to sentence or paragraph. Table 2 shows examples for the different context sizes.

The fourth and crucial element of a WSD process is the selection of the actual classification method itself. In general, three different approaches can be considered: supervised, unsupervised and knowledge-based disambiguation. Supervised WSD algorithms derive statistical models and classification rules mostly from sense-labeled corpora. Unsupervised algorithms deduce divisions of senses from untagged training samples. Sometimes, such approaches make use of external knowledge bases, such as WordNet's concept taxonomy [125]. Knowledge-based algorithms strictly rely on external knowledge sources to distinguish between different senses for different contexts. Related approaches are described in the following paragraphs.

SUPERVISED DISAMBIGUATION Supervised WSD approaches use machine-learning algorithms to train classifiers based on manually annotated corpora. The training set consists of a set of examples in

7 http://en.wikipedia.org/wiki/Jaguar

which the words are manually tagged with one of the senses from the sense dictionary. Several machine learning techniques have been applied for the purpose of WSD, such as Support Vector Machines (SVM) [58], decision trees [91], neural networks [20], or naive Bayes classifier [83].

UNSUPERVISED DISAMBIGUATION Unsupervised WSD does not require a sense-annotated corpora. Thereby, the *knowledge acquisition bottleneck* described in [35] can be overcome. These approaches are based on the assumption that words occur with similar sets of context words when the same sense can be applied. Word occurrences are clustered and new occurrences are assigned to already induced clusters. Obviously, unsupervised WSD does not strive for sense annotation, but rather for sense discrimination by clustering texts containing the same senses [82]. For this purpose, unsupervised WSD approaches can be differentiated into word clustering [14], context clustering [97] and co-occurrence graph methods [26].

KNOWLEDGE-BASED DISAMBIGUATION Supervised disambiguation approaches are dependent on the availability of sense-labeled corpora. The detected senses are restricted to the senses annotated in the utilized corpora. In contrast, knowledge-based approaches are dependent on specific information derived from external knowledge bases, but the knowledge base is interchangeable. Thus, once the algorithm is implemented, it can be applied to any knowledge base compliant with the requirements. Also, knowledge-based approaches achieve a larger coverage due to the availability of large-scale knowledge bases. The *Lesk* algorithm, also called *gloss overlap*, is based on the overlap of sense definitions from the knowledge base and the target context [61]. A more elaborate approach based on the simple *Lesk* algorithm is described in Section 8.2.3 for the proposed semantic analysis process. Structural approaches take into account the structure and semantic network of the senses provided by the knowledge base. Two main approaches can be differentiated: similarity-based and graph-based approaches [82]. For instance, the semantic similarity can be computed from the network of semantic connections between word senses. Rada et al. presented an approach to calculate the shortest distance between the different senses of two words [92]. The senses of the words with minimal distance are considered the most relevant senses for the two words. A different approach has been presented by Mihalcea taking into account the graph structure of WordNet and applying Google's PageRank algorithm for the detection of the most relevant sense within all senses of the words of a given text [76].

APPLICATIONS FOR WSD As a part of NLP, several applications can benefit from WSD approaches. Machine translation algorithms

need to detect correct senses of the words in the source language to translate them into the target language. Content analysis in general, and Information Extraction and Information Retrieval in particular can benefit from the correct disambiguation of the target texts. Navigli also lists the Semantic Web as a real-world application that benefits from WSD [82]. The proposed semantic analysis uses WSD as an essential algorithm to annotate video documents with semantic entities. Section 4.3 introduces more related work on WSD in the research field of the Semantic Web utilizing semantic knowledge bases for the determination of specific senses.

3.2.4 *Coreference Analysis*

Coreference analysis examines text for two or more expressions that refer to the same object. Different specifications are summarized to coreference [53]:

- Anaphora: a proform follows the expression it refers to. Example: **John** went to Atlanta where **he** is attending a conference.
- Cataphora: a proform precedes the expression it refers to. Example: If **he** is on time, **John** won't miss the train.
- Split antecedents: a proform refers to multiple expressions. Example: **John** and **Mary** met in Atlanta. **They** attended the same conference.
- Coreferring nouns: different noun phrases refer to the same object. Example: **Barack Obama** arrived in Berlin. **The US president** is about to meet Angela Merkel.

The general approach to coreference analysis is to keep track of all mentioned entities within a discourse and thereby to create a *discourse model* [54]. An entity-based approach on coreference analysis is introduced in Section 8.2.5 and applied for the proposed disambiguation process.

Recently, Lee et al. introduced an approach to perform rule-based coreference analysis [57]. Rules are applied from highest to lowest precision and each rule builds on the output of the previous rule's output. Similar to the approach presented in Section 8.2, this procedure uses the output of rules with a higher precision as additional input for subsequent processes with presumably lower precision. Output of high precision rules are used as a reference point for the following application of rules. The application of the context model (see Section 7.1) aims to order context information with respect to their confidences and utilizes information with high confidence first within the disambiguation process.

3.3 FURTHER SEMANTIC ANALYSIS METHODS

The presented work applies NLP techniques as introduced so far to detect key terms and named entities in continuous text and to disambiguate them. Moreover, semantic analysis of natural language text is able to inspect the structure of the text and detect relationships among several component parts. The results are used to draw conclusions or to classify the text. The following paragraphs briefly describe further semantic analysis methods.

SENTIMENT ANALYSIS In the course of online marketing, sentiment analysis (also referred to as opinion mining) is an essential part of text analysis. The task aims to detect positive and negative sentiments in (web) documents. Sentiment analysis is useful for companies wishing to track opinions about their products online, as well as for political parties who want to track opinions about specific political strategies. The main challenge in sentiment analysis is the variety of linguistic usage in natural language, including the use of sarcasm. The poor performance of automatic algorithms on the detection of sarcasm is one reason why sentiment analysis still is a challenging research field [70]. A recent approach takes into account entity extraction and the subsequent detection of positive, negative or neutral sentiments with respect to the extracted entities [71].

Sentiment analysis is not part of the proposed work, as the pure presence of specific entities is more important than a detected opinion about them for the purpose of semantic searches of videos (see Section 4.2 for more details on semantic searching).

SHALLOW SEMANTIC PARSING Semantic parsing aims to identify semantic roles held by specific parts of a sentence. Semantic roles can range from rather abstract roles, such as *agent* or *patient*, to more domain-specific roles, such as *speaker*, *message* or *topic* [37]. For instance, consider the following two sentences [116]:

- [The GM-Jaguar pact]$_{\text{AGENT}}$ *gives* [the car market]$_{\text{RECIPIENT}}$ [a much-needed boost]$_{\text{THEME}}$.
- [A much-needed boost]$_{\text{THEME}}$ was *given* to [the car market]$_{\text{RECIPIENT}}$ by [the GM-Jaguar pact]$_{\text{AGENT}}$.

Although the phrases are positioned differently, they are labeled with the same semantic roles with respect to the verb *give*. Mainly, semantic parsing is based on the training of domain-specific corpora. Domain-independent classifiers include rather abstract roles, such as *manner* or *temporal*, as included in the Penn Treebank corpus [37]. Like sentiment analysis, shallow semantic parsing is not part of the semantic analysis presented in this work.

4 LINKED DATA AND THE SEMANTIC WEB

This chapter introduces the basic concepts of the Semantic Web (SW), its application for the purpose of semantic search and semantic analysis.

4.1 SEMANTIC WEB TECHNOLOGIES

The fact-centric SW constitutes an extension of the document-centric WWW. Furthermore, in the Semantic Web the *information is given a well-defined meaning, better enabling computers and people to work in cooperation* [12]. It provides the explicit meaning of the information contained in the documents of the WWW as standardized and structured representations of knowledge. In 2001, Tim Berners-Lee first introduced his new vision of the SW as an extension of the WWW [11]. In his vision, the SW enables computers to understand the content of the web, similar to human understanding. In the course of overcoming the challenges of the Semantic Web, three processes have been identified [49]:

1. building abstract models to describe the world and express human knowledge,
2. computing with knowledge given reasoning machines to derive conclusions from this knowledge, and
3. exchanging complex information to connect knowledge as a global knowledge base.

Abstract models can be built with the help of the Resource Description Framework (RDF). RDF aims to represent knowledge about resources as triples – as assertional knowledge. Conclusions can be drawn from the abstract models in terms of implicit knowledge which is represented by constraints and rules – as terminological knowledge. In the first instance, this is achieved by the development of Resource Description Framework Schema (RDFS). The syntax and semantics of RDF(S)[1] are introduced in Section 4.1.1. More sophisticated constraints are modeled by using the Web Ontology Language (OWL) which is based on Description Logics (DL). OWL is introduced in Section 4.1.2. The third process corresponds to the concept of *Linked Data*, whereby semantic data about resources from multiple domains are published and interlinked through relationships. Linked data and some knowledge bases are introduced in Section 4.1.4.

1 The notation RDF(S) refers to both RDF and RDFS.

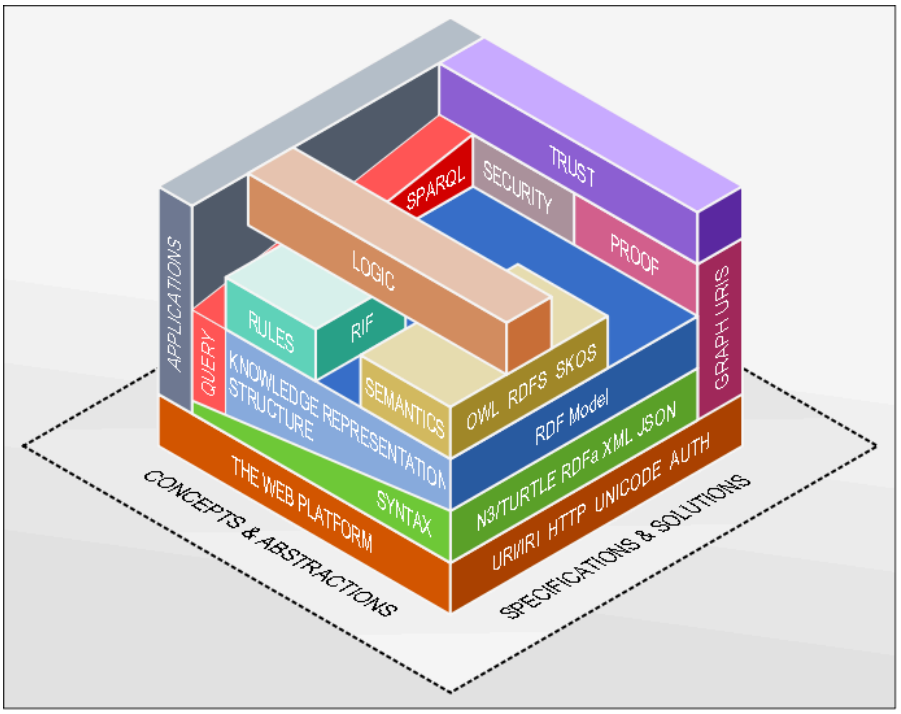


Figure 1: Semantic web stack representing the technologies developed for the Semantic Web.

The technologies developed for the SW are summarized in the Semantic Web stack depicted in Figure 1. A central part of the technology stack is the SPARQL Protocol and RDF Query Language (SPARQL). The language enables the sophisticated query of RDF facts. The technology and some examples are described in Section 4.1.3.

The semantic representation of facts – and even more importantly the semantic annotation of documents – enables a semantic search. Multiple specifications of semantic search have been developed and published. Semantic video search is one of the developed technologies and the main application of the proposed work. Section 4.2 gives and introduction on the main concepts of semantic search.

Section 3.2.3 introduced the concept of WSD for the purpose of assigning specific meanings to common words and named entities. This approach is enhanced and further developed by using semantic knowledge bases as external resources and sense dictionaries. Section 4.3 gives an introduction on annotation services using WSD for automatic annotation of (web) documents.

4.1.1 *Abstract Models with RDF(S)*

RDF is a data model that presents facts in a universal, machine-readable exchange format for the purpose of describing structured information. The information exchange via RDF strives to preserve the original meaning and ensure a common understanding of the data on

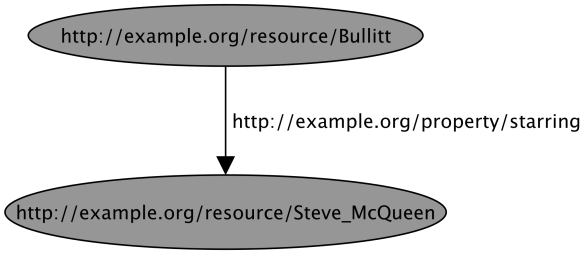


Figure 2: Simple example of a directed graph.

both sides of the exchange – sender and receiver. An RDF document represents a directed graph of knowledge. The graph consists of a set of nodes that are connected by directed edges. Nodes and edges between the nodes are named by unique identifiers – a Uniform Resource Identifier (URI) [49]. Figure 2 shows a simple example of a directed graph.

Such directed graphs can also be represented by triples of `subject`, `property` and `object`. The edges represent the properties and the nodes are transformed to subject and object in the direction of the edge. The graph depicted in Figure 2 can thereby be transformed to the following triple (replacing the prefixes of the URIs with the specified abbreviations):

```
@prefix exr: <http://example.org/resource/> .
@prefix exp: <http://example.org/property/> .
exr:Bullitt exp:starring exr:Steve_McQueen .
```

RDF graphs and thereby RDF triples describe relationships between *resources*. Properties must be represented by a URI. RDF `objects` (the object of an RDF triple) can also be represented by literals. For instance:

```
@prefix exr: <http://example.org/resource/> .
@prefix exp: <http://example.org/property/> .
exr:Steve_McQueen exp:name "Terence Steven McQueen" .
```

Additionally, subjects and objects of triples can also be anonymous blank nodes. Consider the following example:

```
@prefix exr: <http://example.org/resource/> .
@prefix exp: <http://example.org/property/> .
exr:Bullitt exp:starring :_1 .
:_1 exp:name "Terence Steven McQueen" .
:_1 exp:birthDate "1930-03-24" .
```

Blank nodes identify resources within the same RDF document, but cannot be referenced from outside the document.

RDFS has been developed to create simple ontologies[2] in RDF. This enables several enhancements of knowledge represented as RDF:

2 A detailed definition of ontologies in the scope of the SW is given in Section 4.1.2.

- distinguish between classes and instances,
- determine domain and range of properties, and
- define hierarchies of classes and properties.

Classes, such as *person* or *place*, can be defined and individuals can be assigned as instances of these classes. For example:

```
@prefix exr: <http://example.org/resource/> .
@prefix exp: <http://example.org/property/> .
@prefix exo: <http://example.org/ontology/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
exr:Bullitt rdf:type exo:Film .
exr:Steve_McQueen rdf:type exo:Person .
exo:Film rdfs:subClassOf exo:Work .
```

Domains and ranges of properties restrict subjects and objects used with a property to specific classes. For example:

```
@prefix exp: <http://example.org/property/> .
@prefix exo: <http://example.org/ontology/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
exp:name rdfs:domain exo:Person .
exr:Steve_McQueen exp:name "Terence Steven McQueen" .
```

However, this constraint does not lead to conflicts when properties are used with *wrong* subjects or objects, because negation cannot be expressed in RDF(S). Rather, a resource used as a subject in a triple becomes an instance of the class defined as the domain of the triple's property. The same applies for the range of properties. More expressive ontologies can be created using OWL, which is described in the next section.

RDF graphs can be represented in various ways. The first proposed notation of RDF triples is the *N3* notation introduced in 1998 by Tim Berners-Lee [10]. In 2004, the RDF recommendation proposed a simplified version of *N3* as *N-triples* [49]. This serialization has been further enhanced resulting in the not yet standardized RDF syntax *Turtle*. Additionally, RDF facts can be serialized in XML syntax. All RDF examples included in this work are written in *Turtle* syntax.

In November 2013, a W3C Candidate Recommendation for RDF version 1.1 was released. The main changes between this recommendation and the recommendation of 2004 pertain to constraints for literals and language tags. Furthermore, the RDF URI Reference has been replaced by IRIs (Internationallized Resource Identifier) where the special characters <, >, {, }, |, \, ^, ‘, double quote and space must be percent-encoded[3].

3 http://www.w3.org/TR/rdf11-concepts

4.1.2 *Ontologies and OWL*

Within the scope of computer science, an ontology has been defined in 1993 by Thomas R. Gruber [44]:

> *An ontology is an explicit specification of a conceptualization.*

A conceptualization is defined as an abstract view (model) of the world that is required to be presented for a specific purpose. Thereby, the set of objects represented by an ontology might belong to a specific domain and determine the knowledge – the so-called universe of discourse. The knowledge must be defined in a declarative formalism and the meaning of all concepts must be specified.

For the Semantic Web, the mentioned declarative formalism is represented by OWL. The latest version of OWL was released in 2009 as OWL 2. To enable ontology engineers to choose between different levels of expressivity, different *species* of OWL 2 have been developed: OWL Full, OWL DL and OWL Lite. OWL Full provides the highest expressivity, but is undecidable mainly due to the missing enforcement of type separations: individuals, classes and properties can be mixed freely [49]. OWL DL restricts OWL Full in the use of some elements so that the language becomes decidable. OWL Lite represents the least expressive sublanguage of OWL. The following explanations mainly refer to OWL 2 DL.

By using OWL, classes (which represent the concepts), instances of the classes and relationships between instances as well as between classes can be modeled. Accordingly, the following definitions are true for an OWL ontology:

- *Classes* represent sets of individuals sharing certain characteristics, i.e *person*, *country*, *company*.
- *Individuals* are objects of the domain possessing a specific meaning and a unique identifier. Individuals are instances of the ontology's classes.
- *Properties* denote the relationships between the individuals or classes. OWL allows two general types of properties: object and datatype properties. The former represent relations between objects (individuals or classes) of the ontology that are referenced by an URI. The latter relates an object to a data value, i.e. an integer.

OWL 2 DL is a description logic and is considered a semantic fragment of First Order Logic (FOL). It corresponds to the description logic $\mathcal{SHROIQ}(\mathcal{D})$:

- $\mathcal{ALC}$ + transitivity $\rightarrow \mathcal{S}$
- Role hierarchy $\rightarrow \mathcal{H}$

- Reflexivity, irreflexivity and disjointness of properties $\rightarrow$ R
- Nominals, enumerated classes $\rightarrow$ O
- Inverse roles $\rightarrow$ I
- Qualified cardinality restrictions $\rightarrow$ Q
- Use of datatypes $\rightarrow$ (D)

The following class constructors are allowed:

- conjunction $\sqcap$
- disjunction $\sqcup$
- negation $\neg$
- existential quantifier $\exists$
- universal quantifier $\forall$

They can be used to construct and describe classes, such as the class of "flying, grass-eating pigs" FGEP:

$$\mathtt{FGEP} \subseteq \mathtt{FlyingObject} \sqcap \forall \mathtt{eats.Grass} \sqcap \mathtt{Pig}$$

In addition to existential and universal quantifiers, cardinalities can be used to restrict the use of properties to a specific number for instances of a class. For instance, a class of vehicles with four or two wheels can be constructed.

Further reading on ontologies can be found in [104]. The different species of OWL 1, OWL 2 and OWL syntax are further described in [49].

4.1.3 *Query of Facts via SPARQL*

SPARQL enables a query of RDF-based information using graph patterns. Additionally, it provides further functions for filtering conditions, advanced query patterns, and formatting the output [49].

SPARQL queries are represented in Turtle syntax (see Section 4.1.1). Unknown nodes in the query graph are replaced by variables. These variables can be used for further filtering and to construct the output. Thereby, complex queries on RDF graphs are enabled. For instance, a fictional SPARQL query for all movies set in New York City, released before 1970, and starring Audrey Hepburn might look like the following:

```
PREFIX exr: <http://example.org/resource/>
PREFIX exp: <http://example.org/property/>
PREFIX exo: <http://example.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?film WHERE {
?film rdf:type exo:Film .
?film exp:starring exr:Audrey_Hepburn .
?film exp:filmLocation exr:New_York_City .
?film exp:publicationYear ?year .
FILTER(?year<1970)
}
```

An enhanced version of SPARQL became a W3C recommendation in March 2013. Amongst others, the new version SPARQL 1.1[4] enables property paths, property disjunction, grouping functions and aggregate functions.

4.1.4 *Exchange Information – Linked Data Cloud*

In 2006, Tim Berners-Lee published the *Linked Data Principles*[5]. They have been established to allow exploration of the web of data provided by the SW:

- Use URIs as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using standards (RDF, SPARQL).
- Include links to other URIs, so that users can discover more things.

In the course of recent years, many datasets have been published as Linked Data. The *Linking Open Data* project states its goals as *publishing various open data sets as RDF on the Web and at setting RDF links between data items from different data sources*[6]. The resulting linked datasets are called the *Linked Open Data* (LOD) cloud. Figure 3 shows a snapshot of the cloud as of September 2011. As of November 2013, the cloud consists of 62 billion triples from 870 datasets. The statistics of the LOD cloud are maintained by the LODstats project[7].

DBPEDIA The DBpedia[8] represents a link hub within the cloud, as many LOD datasets link their entities to DBpedia entities. The DBpedia is the semantic version of Wikipedia[9]. Each Wikipedia article represents a DBpedia entity. The entity types are derived from the infobox templates used for the Wikipedia article and facts about the entities are derived from the information linked in the infoboxes. Additionally, information about redirects, disambiguation pages, and page

4 http://www.w3.org/TR/sparql11-query/
5 http://www.w3.org/DesignIssues/LinkedData.html
6 http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
7 http://stats.lod2.eu
8 http://dbpedia.org
9 http://www.wikipedia.org

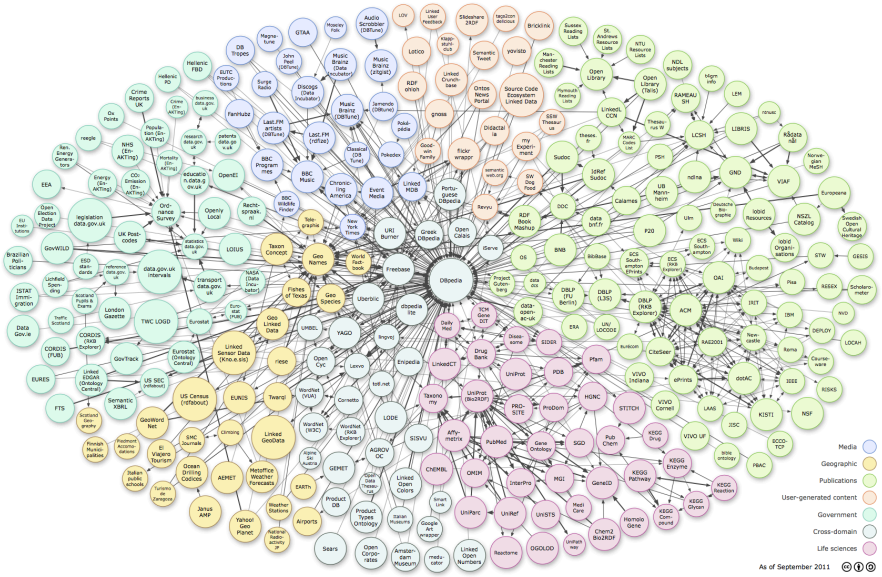


Figure 3: Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch.
http://lod-cloud.net/

links are extracted as RDF triples. The latest version of the English DBpedia[10] consists of 2.46 billion facts about (approximately) 4 million things/objects. The DBpedia dataset uses an original ontology that has been created manually based on the most commonly applied infoboxes. As of its latest version (in conformity with the dataset itself), the ontology comprises 256 classes and 2,333 properties. The properties and classes are derived from the infoboxes by a crowd-sourced approach. A public wiki is used to map new infoboxes and properties used in infoboxes to existing classes and properties respectively, create new ones and integrate them in the ontology. The ontology mainly consists of basic class definitions comprising class name labels in several languages and property definitions including domain and range (but only for a subset of all properties). It does not contain strict class constraints, such as composed classes or cardinality restrictions. Several research approaches deal with the detection of inconsistencies in the DBpedia ontology, including the dataset. Mostly these approaches aim to compare ontology definitions with (contradictive) facts in the dataset [114, 64].

GND In 2011, the authority files of the German-speaking countries were published by the German National Library (DNB) under the name *Gemeinsame Normdatei (GND)* as Linked Data. The dataset originates from the German library community and contains content of the former Corporate Body Authority File (GKD for Gemeinsame Körperschaftsdatei), the Name Authority File (PND for Personenna-

10 Version 3.9 as of March/April 2013

mendatei), the Subject Headings Authority File (SWD for Schlagwortnormdatei) and the Uniform Title File of the Deutsches Musikarchiv (EST for Einheitssachtitel-Datei). As of November 2013, the GND dataset contains over 10 million RDF triples and maintains links to the datasets *Library of Congress Subject Headings (LCSH)* and the *Virtual International Authority File (VIAF)*, amongst others. The GND dataset also provides its own completely manually created ontology. It comprises 50 classes, 162 object properties and 56 datatype properties. Domains and ranges are defined for all properties. Additionally, as the only axoim a list of disjoint classes is provided (*ConferenceOrEvent*, *CorporateBody*, *Family*, *Person*, *PlaceOrGeographicName*, *SubjectHeading*, *Work*).

Both datasets, DBpedia and GND, have been utilized to create knowledge bases for the semantic analysis of video metadata presented in this work. More details on the creation process of the knowledge bases are described in Section 8.1.

Semantic Web technologies enable new search approaches taking into account the meaning of a search query. Several approaches of *semantic search* are introduced in the next section.

4.2 SEMANTIC SEARCH

Typically, keyword-based search algorithms return an ordered list of (web) documents containing the string entered by the user. More sophisticated approaches also take into account synonyms of the search input. Still, the algorithms are limited to a simple text-based search without taking into account the semantics of the search input. The task of semantic search can be interpreted in two different ways:

- Understand and interpret the search input and provide an answer.
- Handle entity-centric search input and provide documents annotated with the demanded entities.

An example for the first approach is the search engine WolframAlpha[11]. It can handle questions like *How long is the river Rhine?* and returns *1230 km* (see Figure 4). WolframAlpha is based on the computational software *Mathematica*[12]. The input question is analyzed and interpreted and the answer is calculated based on the information provided by hundreds of datasets [122].

The second type of semantic search requires a corpus of annotated documents. The documents can either be annotated with relevant categories or subjects, or the content can be analyzed for containing entities. This type of semantic search seeks to increase the accuracy

11 http://www.wolframalpha.com
12 http://www.wolfram.com/mathematica/

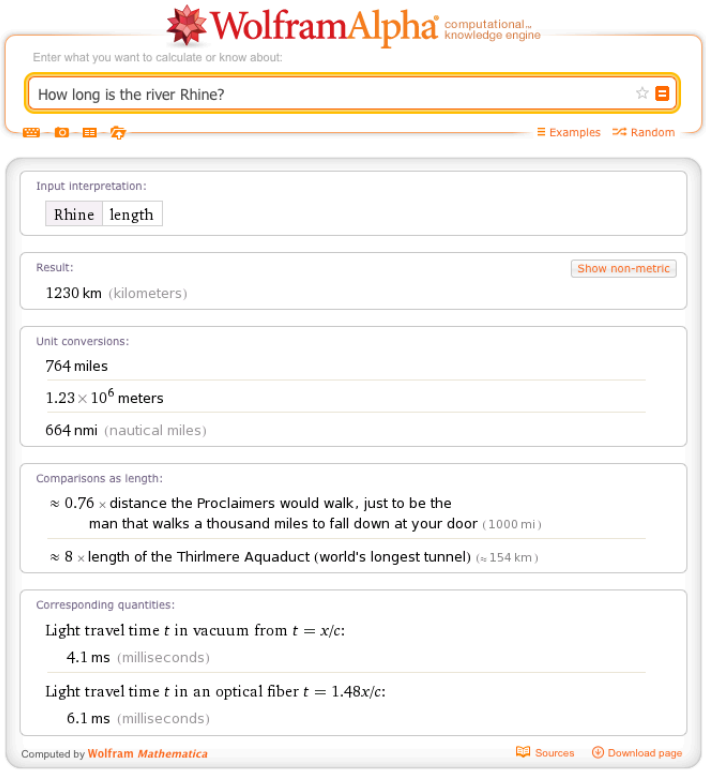


Figure 4: Search result of WolframAlpha for the question *How long is the river Rhine?*

of search results. Keyword-based search algorithms often provide results that are not relevant for the search input. Although the resulting documents may contain the search input (or a synonym), they might possess a different meaning than the input. For instance, a Google[13] search for the input string *jaguar* returns documents both about the car manufacturer and the feline animal.

In contrast, an entity-centric semantic search asks the user for the specific entity she is interested in and then returns only documents annotated with this entity (possessing the intended meaning). Furthermore, the relationships between the annotated entities of different documents can be utilized to provide an *explorative search* within the underlying document archive [120]. Figure 5 shows the exploratory search GUI of Yovisto[14] for the search input *American president*. The search result provides videos about American presidents. Additionally, related entities are provided (on the left) to start an exploratory search through the video archive.

This type of semantic search constitutes one of the applications utilizing the results of the proposed work of semantic analysis of video metadata. More details on current applications and search portals can be found in Chapter 11.

13 http://www.google.com
14 http://www.yovisto.com

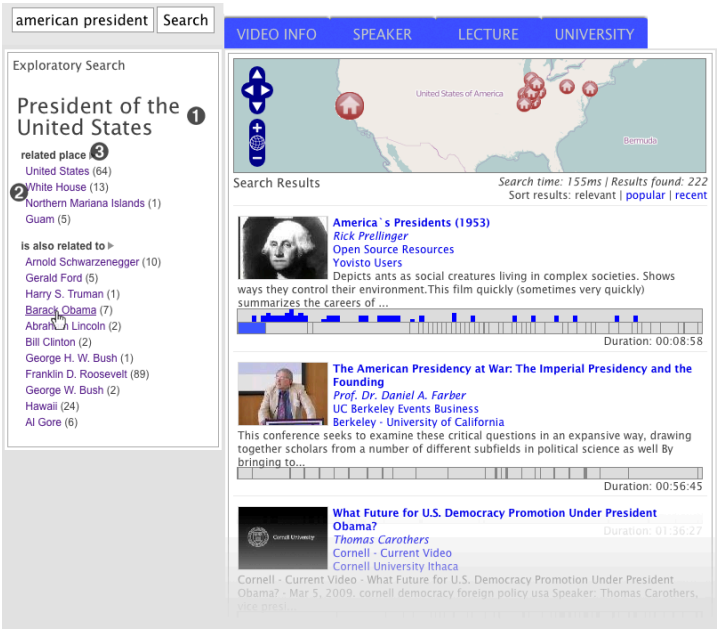


Figure 5: Search result of Yovisto for the input *American president*. The input is disambiguated to *President of the United States* and videos are provided annotated with this entity. On the left, related entities are provided for an exploratory search.

4.3 AUTOMATIC SEMANTIC ANNOTATION

The preceding section introduced semantic search on annotated (web) documents. Manual annotation of documents is time-consuming and often requires domain experts. Automatic annotation tools provide the analyzed document with the most relevant annotations according to the context given by the document itself. Similar to WSD introduced in Section 3.2.3, natural language text is analyzed and ambiguous terms are disambiguated. Within the scope of the SW, the texts are annotated with semantic entities of an underlying Linked Data dataset and disambiguated using the semantic information provided for the entities within (and across) the dataset. In recent years, several annotation algorithms have emerged and respective tools have been published. Most of the approaches link spotted terms to Wikipedia articles (DBpedia entities). Wikipedia constitutes a trade-off between well structured, manually maintained catalogs, such as WordNet or Cyc[15], which feature a low coverage of entities and common terms and a large but noisy collection of texts (such as the web) [46].

The following section introduces some of the first approaches to semantic annotation, as well as current state-of-the-art tools linking to Wikipedia. The section is concluded with a discussion of the approaches and an argumentation for the proposed work.

15 http://www.cyc.com

In terms of the spotting algorithm, two different annotation approaches must be distinguished: Named Entity Disambiguation (NED) and WSD. The first approach only takes into account named entities, which means that NER is used to identify the names of persons, places or organizations in the text and only these parts of the text are annotated with the respective entity. The latter approach annotates and disambiguates both named entities and common nouns in the text.

Mihalcea et al. have published one of the first WSD approaches using Wikipedia articles to identify specific entities [77]. The authors propose a combined approach of an analytical method comparing Wikipedia articles with contextual paragraphs and a machine-learning approach for the disambiguation process.

Another WSD approach based on a trained classifier is presented in [18]. The approach uses specific kernels in linear combination to disambiguate terms in a given text. The kernels are trained with surrounding words of an entity link within the paragraphs of the Wikipedia article. The online annotation service *Wiki Machine* corresponds to the approach presented in [18][16].

DBpedia Spotlight is an application that detects and disambiguates both named entities and common words in continuous text. The context information of the text to be annotated is represented by a vector. Every entity candidate of a term[17] found in the text is represented as a vector composed of all terms that co-occur within the same paragraphs of the Wikipedia articles where this entity is linked [75].

Recently, Damljanovic et al. presented an approach of combining NER tagging and WSD [22]. The terms spotted and identified by the NER tagging tool (person, place or organization) are assigned to DBpedia classes. Entity candidates for the spotted term are retrieved only within the instances of the assigned ontology class.

The goal of *TagMe* is to pick out common words and named entities particularly in very short texts [30]. Their approach builds upon the graph-based approach introduced by Milne and Witten [78]. It takes into account the Wikipedia page link graph and the relationships of the entity candidates to all terms spotted in the input text. *TagMe* has been enhanced to utilize this approach in very short texts, such as Twitter[18] posts.

AIDA is an online tool for the disambiguation of named entities in natural language text and tables [126][19]. It utilizes relationships between named entities for the disambiguation. Plain text, HTML as well as semi-structured data, such as tables, are accepted as input formats.

16 The service is no longer available online.

17 The authors use the expression *surface form* for a word or a word group representing an entity. Throughout the present work *term* is used synonymously to this definition.

18 `http://twitter.com`

19 Therefore it is considered an NED approach.

Zemanta[20] and OpenCalais[21] are annotation tools that provide semantic annotations for an input text. In comparison to the previously introduced approaches, both tools only sparsely identify named entities but focus on recommending web documents or web sites for the given text.

DISCUSSION All of the referenced annotation approaches aim to analyze documents containing continuous text. Context definitions are limited to merely structural classes such as word, sentence, paragraph or full document [99]. All context information is treated equally with respect to provenance and other characteristics. This procedure results from the assumption that continuous text can be considered as an homogenous context. However, this assumption cannot be applied to video metadata or other (web) documents possessing metadata from different sources and with different characteristics. Chapter 6 introduces definitions of context applicable for such documents and an annotation algorithm developed for the handling of heterogenous contexts with respect to the characteristics of the information pertaining to it. Chapter 10 presents an evaluation of the proposed approach against four of the previously introduced annotation tools: *DBpedia Spotlight*, *AIDA*, *TagMe* and *Wiki Machine*.

20 http://www.zemanta.com
21 http://www.opencalais.com

5

DEFINITIONS OF CONTEXT

The discussions about the influence and importance of context date far back throughout various fields of computer science. In 1931, John Dewey wrote "We grasp the meaning of what is said in our language not because appreciation of context is unnecessary but because context is inescapably present." [23] Although this sentence addresses context in the field of psychology, it is also valid for the context considered for computer science. This chapter gives an overview of context definitions as applied in different research fields such as linguistics, mobile computing and cognitive sciences. The concept of *negative context* is introduced in Section 5.2 as it is part of the proposed semantic analysis process. Context, pragmatics and semantics are closely interrelated, and Section 5.3 examines these interdependencies.

5.1 CONTEXT IN DIFFERENT RESEARCH FIELDS

In 1884, Gottlob Frege phrased three fundamental principles on general logic [34]:

- always sharply separate the psychological from the logical, the subjective from the objective;
- never ask for the meaning of a word in isolation, but only in the **context** of a proposition; and
- never lose sight of the distinction between concept and object.

Thereby, Frege introduced the *context principle* which underscores the important issue of context for the interpretation of textual information.

COMPUTER SCIENCE - MOBILE COMPUTING Context and context-aware computing have received increasing attention, in recent years [74]. In e-commerce and ubiquitous computing, context is applied as the context which encloses a person (as a user). Therefore, characteristics of context are defined to solve personalization problems, to identify life stages of a person for data mining, or to improve online marketing and management [3]. This context can be considered as user context. In 1999, Schmidt et al. developed a context model to represent the characteristics of a user context [95]. The model combines human factors (user, social environment, task), and physical environment factors (conditions, infrastructure, location). In 2000, Abowd and Mynatt proposed five questions (the *Five Ws*) answering necessary questions about user context [1]:

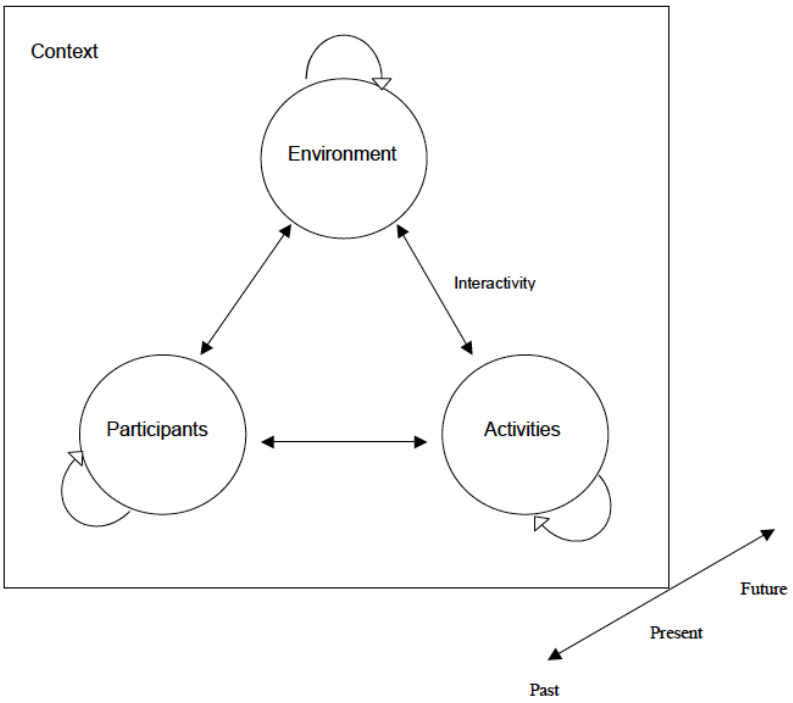


Figure 6: Context model describing user context introduced in [112].

- Who – the user and other people in the environment
- What – human activity perception and interpretation
- Where – location and the perceived path of the user
- When – time as an index and elapsed time
- Why – reason a person is doing something

These approaches were further enhanced in 2003 by Tarasewich [112]. The model defines three broad categories of context: environment, participants and activities (see Figure 6). This tripartite breakdown emphasizes the fact that beside the user herself, others can be part of a given context. Additionally, interactions and relationships between participants, activities, and the environment are included in the model. A recent approach takes into account some aspects of the context models previously introduced and links them to the application's context [42]. Thus, factors relevant to the different participants as well as perspectives of a research and development process are taken into account.

LINGUISTICS In the field of linguistics, a generally accepted truism is that an utterance is dependent upon its (social, situative and sequential) context. This dependency of natural language and its semantic meaning on the given context has been described in several theories. Extra-linguistic reference points which contain information

about the semantic interpretation have been introduced to prove this assumption [6]. In 1970, Lewis defined six *contextual coordinates corresponding to familiar sorts of dependence on features of context*[62]. Thereby, a *time coordinate*, *place coordinate*, *speaker coordinate*, *audience coordinate*, *indicated-objects coordinate*, and a *previous discourse coordinate* have been introduced to describe a linguistic context.

Auer characterizes a context applied in linguistic research as follows [6]:

- Context is considered as aggregate of the given independent entities. The entities exist independent from interactions occurring within the context.
- The awareness of context is assumed. Divergences are not under consideration any more than interactive problems resulting from a lack of knowledge.
- The effect of context on the interaction is uni-directional. Context influences the linguistic behavior but not vice-versa.

For conversations, Akmajian et al. state that context is affected as well as reflected by contributions to a discourse. Thus, the context might change by reflecting a previous context. Or the context might be affected and specified by previous conversations [4]. Similar to this theory, Dijk emphasizes the *dynamic* character of context. Hence, a context is a *course of events* possessing an initial state, (several) intermediary states and a final state [118]. However, Dijk also states that a context must have limits. The conditions of a possible world need to be known *in order to qualify as an initial or final state of context*.

SOCIO-LINGUISTICS AND COGNITIVE SCIENCES The term *contextualization* was introduced for the first time by John Gumperz and Jenny Cook-Gumperz [45] in 1976. Contextualization specifies an active interaction participant who not only reacts but also creates context. Interaction participants are suggested to perform linguistic activities and allow them to be interpreted by creating contextual information. This hypothesis provides the foundations for considerations on context in socio-linguistics and cognitive sciences. Although there is agreement on the difficulties of defining context in general and finding a universal definition, the different disciplines identify certain characteristics within their fields of interest. Lenat [60] states that for artificial intelligence, context has been ignored or treated as a black box for a long time. For the popular knowledge base Cyc[1] he defined twelve dimensions of context to "specify the proper context in which an assertion (or question) should be stated". Bazire et al. collected 150 different definitions of context from various disciplines of cognitive sciences to identify the main components of context [8]. The study

1 http://cyc.com/cyc/opencyc

is concluded by the determination of all definitions for the parameters constraint, influence, behavior, nature, structure and system. In ubiquitous computing, context is broadly used for two purposes: as a retrieval cue and to tailor the behavior and the response type of the system [27]. Dourish further identified two different views on context, a representational and an interactional view, and suggests the latter is the more challenging for the field of interactive systems.

5.2 NEGATIVE CONTEXT

Context might influence the interpretation of an utterance in different ways. The common approach takes context information into account to achieve a positive influence on a specific interpretation (*positive context*). The notion of *negative context* is to enable negative influence on specific interpretations and be able to exclude them for a given context.

The concept of negative context is novel for the research fields of semantic analysis and WSD. However, context and negative context are concepts used in other research fields, such as linguistics, psychology and sentiment analysis.

NLP defines so-called negative and positive polarity items that bind a specific context. Negative polarity items (NPI) are lexical elements that occur in negative contexts only, such as the term *any*, while positive polarity items are in general excluded from negative contexts such as the term *some*. Thus, the definition of a negative context depends upon the language specific characteristics and designates contexts that license the occurrence of undisputed NPIs [117].

Negative emotional contexts denote situations where people feel unhappy with the result of an event [25]. Sentiment analysis deals with the problem of negation or agreement within natural language to find out positive or negative influences in a context [89].

Deep or conceptual semantic analysis of natural language also addresses negative context effects [73]. Various contextual situations influence the comprehension of ambiguity in natural language. McCrae describes cross-modal contexts to classify different levels of cognitive input. The presented work raises the issue of different contexts to the Linked Data level and addresses negative context in terms of semantic analysis approaches.

Van Dijk described *negative actions* for the theory of linguistic actions. Hereby, the context is changed by so-called *non-doings* [118].

Machine learning uses positive but also negative training samples to create a classifier [79]. By providing negative training examples, the classifier learns which concept is not meant to be classified and should therefore be excluded from positive classification. For purely analytical approaches, this cannot be achieved in the first place. Sec-

tion 9.2.2 introduces an approach on the derivation of negative context information dynamically.

5.3 SYNTAX, PRAGMATICS, AND SEMANTICS

WordNet[2] lists two different meanings for the word *context*. One meaning lists *linguistics* and *context of use* as synonyms, while the other meaning lists *circumstance* and *setting*. Both interpretations depend on three criteria for distinguishing the multiple functions of context: syntax, pragmatics and semantics [101].

Charles Morris proposed the relationship of the three concepts syntax, pragmatics and semantics in 1938 [81]. Syntax deals with the formal relationships of signs to each other without taking into account their meaning. Semantics examines the relationships between signs and the objects they reference independent of the way the signs are used. Pragmatics, in turn, deals with the relation of signs to their interpreters and is defined as the study of language use [121].

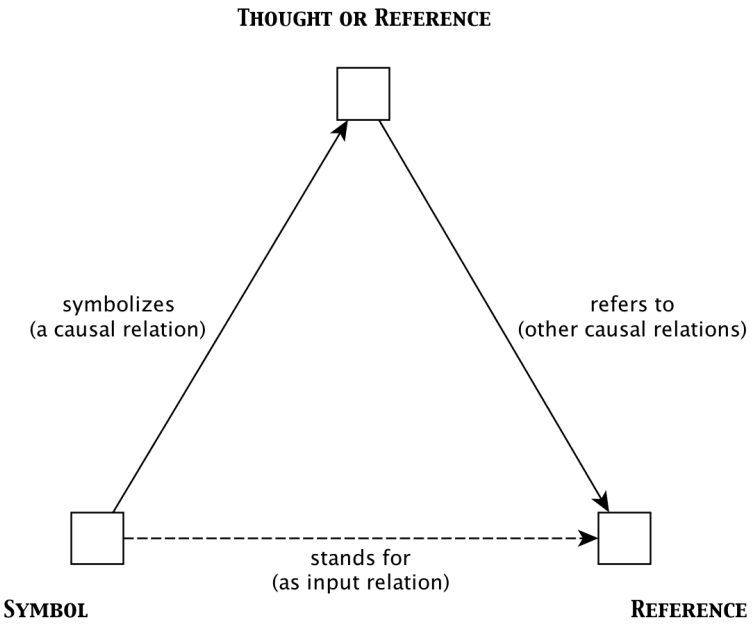


Figure 7: Semiotic triangle according to Ogden and Richards [87].

The union of these three notions is often referred to as *semiotics*. One of the key concepts of semiotics is the *semiotic triangle* (see Figure 7). Philosphical discussions about symbols of objects and references date as far back as Aristotle, but in recent literature the triangle is often referenced as the Ogden-Richards triangle referring to Charles K. Ogden and I.A. Richards as the most popular ambassadors [87].

2 http://wordnet.princeton.edu

It demonstrates the relationship of a linguistic symbol that creates a specific interpretation (reference) in a person's mind and stands for a real object or concept (referent). The semiotic triangle is also a key concept of the Semantic Web and facts represented by URIs in triples.

Thus, a context-aware semantic analysis requires consideration of syntax, pragmatics and semantics. The following chapters introduce definitions and characteristics of context for this purpose. The context description model considers syntax, semantics, and pragmatics to describe heterogenous contexts of video metadata. Hence, all criteria of contexts are combined in the proposed semantic analysis process.

Part II

SEMANTIC ANALYSIS FOR HETEROGENOUS CONTEXTS

6 CONTEXT

This chapter introduces the definition and characteristics of context as required for the semantic analysis of video metadata. Section 6.1 introduces the definition of context for the proposed approach. The size of a context defines its specificity and is limited by its boundaries. Context boundaries are defined differently for distinct document types. Section 6.2 introduces different identified context boundary characteristics. A context is given naturally, but might be influenced by manual or automatic refinement. Context refinement strategies are presented in Section 6.3. Information provided in a context might be used as supporting information for a specific interpretation; this type of information is subsequently referred to as *positive context*. Likewise, context information might be used to contradict a specific interpretation. Consequently, this type of information is subsequently referred to as *negative context*. These specifications of context are presented in detail in Sections 6.4.1 and 6.4.2. Metadata information of multimedia documents originates from multiple sources and possesses diverse characteristics. Thus, contexts consisting of metadata information of multimedia documents, such as videos, is considered heterogenous. Characteristics of heterogenous contexts and respective examples are introduced in Section 6.5.

6.1 CONTEXT IN SEMANTIC ANALYSIS

Context is a term that must be considered theory-dependent [43]. For the proposed approach the definition of *context* must consider aspects of linguistics as well as the Semantic Web. Thus, context is defined as follows:

Definition 6.1.1. A context is a discourse of information that supports and influences its semantic interpretation.

Following this definition, a context can be characterized by its

- granularity,
- type of influence (positive or negative), and
- structure.

Granularity refers to the amount of information within a context and is dependent upon the context boundaries (see Section 6.2). The common definition of a context refers to a positive context, whereby the information pertaining to a context influences the interpretation in a

positive manner (see Section 6.4.1). A negative context refers to information that contradicts a specific interpretation of the given context (see Section 6.4.2). The information pertaining to a context can be of different characteristics. Therefore, the context structure is considered either homogenous or heterogenous (see Section 6.5).

These aspects of context are discussed in the following sections.

6.2 CONTEXT BOUNDARIES

A context boundary demarcates two or more contexts of separate content and influence on semantic interpretation. Context boundaries restrict the amount of information pertaining to a context. Moreover, information between context boundaries – and therefore pertaining to the same context – might be considered coherent with respect to the topic. Here, a topic is considered to be a self-contained subject or chunk of knowledge. The amount of information provided within a context is essential for the correct interpretation. A context which is too broad and hence contains too much information might result in a scope which is too wide for the correct interpretation. On the other hand, too little information might preclude interpretation of the context. Either case might result in a wrong interpretation. The extent of a context can be defined by structural, temporal or spatial levels – depending on the considered document type. All types of context boundaries aim at the same result: division of a given amount of information into coherent parts with respect to the topic of the content. The division obtained by the context boundaries enables a relevance view on information items regarding the appropriate context. This relevance view is discussed in detail in Section 7.3.2. The following paragraphs give a detailed overview of the different context boundary types – structural, spatial, and temporal – for multimedia documents.

6.2.1 *Structural Boundaries of Natural Language Text*

Natural language text within text documents is typically structured in sentences, paragraphs, sections, and chapters. These structural boundaries limit the textual information pertaining to the same context. While some natural language texts might provide a context as a whole, other documents are heterogenous, wherein various subjects and contexts change within sections. For instance, the context might remain the same for an entire document about a very specific medical disease pattern. Meanwhile, a news article about medical innovations might report about various topics and its contexts might change after every paragraph (see Section 3.2.3 for more about context sizes in the research field of WSD in continuous text).

Sample contexts for DBpedia entities can be represented by structural parts of Wikipedia articles containing links to the entities. How-

Figure 8: Spatial boundaries of different context information

ever, DBpedia Spotlight and AIDA utilize sections of Wikipedia articles as sample contexts for the disambiguation progress (see Section 4.3 for a description of the approaches).

Besides the fixed determination of context boundaries, they might also be detected individually for every text. Fort et al. presented an approach to the identification of textual contexts by using a machine learning model on syntactic features [32].

6.2.2 *Spatial Boundaries of Multidimensional Information*

Spatial boundaries define the size of a context by means of the spatial distance to a specific center point. For example, visual information in pictures can be very heterogenous. By dividing a picture into spatial subdivisions, distinct contexts can be defined. Similarly, the zone of interest of a city can be drawn by a line around it on a geographical map – possibly by using district or country borders.

Textual information in broadcast news stories displayed to support the understanding of the content often has a spatial reference region. Names of displayed persons are overlayed in the picture. But mostly, this textual information is only relevant for the person whose image is displayed directly above the text (see Figure 8). Therefore, the context information is restricted by spatial boundaries.

6.2.3 *Temporal Boundaries of Time-Based Documents*

Context changes within multimedia documents, such as audio and video files, emerge along the chronological sequence of the document.

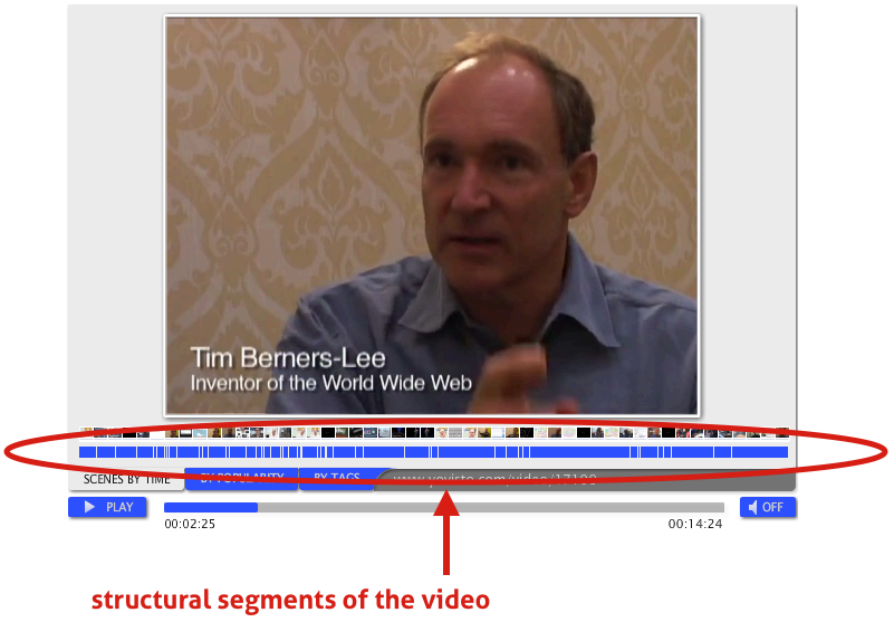


Figure 9: Yovisto video player displaying structural segments of a video

The detection of content-based segments (see Section 2.4.1 for structural segmentation of video documents) results in coherent parts with respect to the content. For instance, a news broadcast covers different stories with various topics. By detecting scene cuts between the news pieces, the temporal context boundaries can be determined.

Figure 9 shows a screenshot of the Yovisto[1] video player displaying a video, its structural segments, and preview thumbnails for every segment. Every segment represents a separate context.

This work focuses on video metadata and therefore temporal context boundaries of video documents. An evaluation of different granularity levels of a temporal context is described in Section 10.7.5.

6.3 CONTEXT REFINEMENT

Independent of Definition 6.1.1, a context can be naturally present but also manually or automatically enabled. Therefore, a context can be refined according to the context's purpose and application. The main purpose of context refinement is the enhancement of an automatic interpretation of textual information. Preexisting knowledge is used to adapt the context. Thereby, the context can be expanded or restricted for the considered purpose. Information is thus determined as permitted or prohibited by the context.

For the proposed work, context is represented by textual as well as semantic information, particularly semantic entities, classes, or categories. Therefore, context refinement mainly considers restricting

1 http://www.yovisto.com

the utilized knowledge base in terms of individual entities or entire classes or categories. On the one hand, the refinement can be handled by allowing a specific list of entities, classes, or categories – whitelists are created. Whitelists define the part of a knowledge base that is permitted for the utilization of context information. On the other hand, entities, classes, or categories that are not allowed can be aggregated in blacklists. Blacklists constitute parts of the knowledge base that are prohibited as context information for a specific scenario. Both approaches are described in detail in the following sections.

6.3.1 *Whitelists vs. Blacklists*

Both, whitelists and blacklists are created by either choosing individual entities or by utilizing properties and types of entities. They make use of aggregated lists of entities.

Whitelists contain accepted entities and entity aggregations with respect to the present context. The information pertaining to a whitelist is permitted as context information utilized for the considered process.

Entities and lists of entities pertaining to blacklists should not be considered for the present context and constitute banned or *negative* entities. This information is prohibited as context information for the considered process. Blacklists also might help to give hints for contradicting facts. This issue is further discussed in Section 6.4.2 with the introduction of *negative context*. Manual selection of individual entities for the creation of whitelists or blacklists requires great effort in either case, especially for knowledge bases containing millions of entities, such as the English DBpedia. Therefore, type or topic information of entities enable aggregation of entities for whitelists and blacklists. The following section introduces specific aggregation approaches for that purpose.

6.3.2 *Aggregation Approaches for Whitelists and Blacklists*

Aggregation of semantic entities according to specific characteristics helps to create whitelists and blacklists. For this purpose, four different characteristics have been identified: type, category, topic, and time reference. These characteristics are described in the following paragraphs in terms of the aggregation of semantic entities.

AGGREGATION BY TYPE Type information of entities is derived from `rdf:type` triples. `rdf:type` is a property "that is used to state that a resource is an instance of a class"[2]. Furthermore, a class provides "an abstraction mechanism for grouping resources with simi-

2 `http://www.w3.org/TR/rdf-schema`

lar characteristics."[3]. The DBpedia ontology provides rather general classes such as persons, places, organizations (and subclasses of the mentioned classes). Thereby, entities with similar characteristics can be aggregated and added to whitelists and blacklists by making use of their type information.

AGGREGATION BY CATEGORY Wikipedia provides several categories that can be assigned to articles (respectively entities). A category mostly aggregates entities that are similar in more than one characteristic. For instance, the category *1916 births* suggests an entity of type *Person* and a birth year of 1916. Likewise, the category *Films set in New York City* suggests entities of type *Film* that are set in the city of New York. Therefore, the application of categories enables an aggregation of entities with multiple similar characteristics.

AGGREGATION BY TOPIC Although categories aggregate entities that are similar in multiple characteristics, they do not represent entire topics, such as *music business* or *sports*. Topics aggregate entities of different types that belong to the same domain. For instance, the topic *Soccer* aggregates entities of type *Soccer Player*, *Stadium*, *Soccer Manager*, *Football League*, etc. The authority file GND (see Section 4.1.4) provides topic information via the property `gnd:SubjectCategory`. Thereby, topics such as *Literature*, *Biology*, *History*, or *Agriculture* are assigned to the entities. The DBpedia does not provide topic aggregations for the contained entities. Topics might be deduced by analyzing the RDF graph. Such an approach has been introduced by Boehm et al. [13]. The approach analyzes used properties and assigned categories of entities and clusters them into topics according to similar characteristics.

TIME REFERENCE AGGREGATION Lots of named entities are time referenced. Persons have a birth date and death date. Places have been founded, renamed or destroyed. Therefore, a context might imply a specific period of time relevant for the mentioned entities. For example, the following two paragraphs both mention *The Prince of Wales*:

- *The Prince of Wales served on the Somme as a staff officer. He was appointed to the staff of Field Marshal Sir John French at the British Expeditionary Force's General Headquarters in France in November 1914.*[4]

3 `http://www.w3.org/TR/owl-ref`

4 Excerpt taken from `www.iwm.org.uk/server/show/nav.2208`.

- *The Prince of Wales is an accomplished horseman and in the 1980s rode in a number of competitive races. He made his debut as a jockey in 1980 at a charity race at Plumpton, East Sussex.*[5]

The first paragraph mentions `dbp:Edward_VIII` who was Prince of Wales from 1910 until 1936 when he became King of England. The second paragraph mentions the current Prince of Wales[6] – and therefore heir to the throne – `dbp:Charles,_Prince_of_Wales` who has been created Prince of Wales in 1958. A whitelist for the first paragraph contains entities that have been born, founded, or created before 1914. A blacklist contains entities that posses their first time reference after 1914, because the latest time reference in the text is the year 1914 and entities with a time reference starting after this year cannot be relevant for the interpretation of the text. Thereby, whitelists and blacklists are created by aggregating named entities according to their and the context's time reference. However, for many named entities a time reference is not given explicitly – at least not in a manner understood by a machine. Whereas persons, places, or companies may possess properties providing time reference information (birth date, founding year, death date), other entities, such as technical inventions (e. g. `dbp:DDR_SDRAM`), do not possess an explicit time reference by nature, but certainly become relevant for an interpretation after their development and first appearance in broadcast media. Amongst others, the research field of *Named Entity Evolution Recognition* deals with this problem and the change of meaning of entities over time (see [111]).

6.4 POSITIVE VS. NEGATIVE CONTEXT

Context information is essential to make a decision about the interpretation of ambiguous text. Most disambiguation approaches (see Section 4.3) take into account context information to find the most supported interpretation within the given context. WSD and NED in the research field of the Semantic Web must deal with the open world assumption. The given context is considered as an excerpt of information that can be taken into account for the correct interpretation. The context cannot be claimed to be complete. Additionally, information *not* given in the context cannot be automatically considered as incorrect or irrelevant. Therefore, a disambiguation process utilizing a given context might be considered as an additive process when the information is used to support the detection of the correct interpretation respective of the most relevant interpretation within the given context. On the other hand, disambiguation can also be organized as a subtractive process. In this way, incorrect interpretations might be identified by means of information *not* relevant for the given context.

5 Excerpt taken from `www.princeofwales.gov.uk/the-prince-of-wales/interests/sports`.

6 As of November 2013.

The additive disambiguation process requires a positive context. The subtractive disambiguation requires a negative context. But, due to the open world assumption, missing context information cannot be considered as negative context. Negative context must be either created manually or deduced automatically. According to this assumption, positive and negative contexts are defined in the following sections.

6.4.1 *Positive Context*

A positive context is determined by its capacity to influence the interpretation of textual information in a supportive manner. Thus, the positive context is utilized to substantiate the decision for a specific interpretation. The more contextual information supports an interpretation the higher is its relevance within the context. The positive context influences the relevance in an additive manner. For example, the term *Berlin* is considered to be interpreted; that is several named entities must be considered for the interpretation of the term. Only taking into account places, there are 29 different entities found within the DBpedia associated with the label *Berlin*. To interpret this term appropriately further context is needed. In this case, the term occurs together with the following terms: *Barack Obama*, *Angela Merkel*, and *Brandenburg Gate*. Therefore, the discourse respective of the positive context for the interpretation of *Berlin* is defined by these three terms. The term *Barack Obama* supports the interpretation of *Berlin* to be one of the towns in the United States with the name Berlin, such as Berlin in Connecticut. But *Barack Obama* also visited the German capital Berlin several times. The other two terms, *Angela Merkel* and *Brandenburg Gate*, support the interpretation of *Berlin* as the German capital. Therefore, the decision for this particular interpretation is supported by all three context terms and is considered more relevant than the other interpretations. The disambiguation process is organized in an additive manner utilizing the terms co-occurring within the same context. Therefore, textual information co-occurring together in a context are considered as positive context. A positive context can also contain entities, topics, categories or type information. However, the information pertaining to a positive context is exclusively used to perform an additive disambiguation process.

6.4.2 *Negative Context*

In contrast to positive context, negative context is utilized to invalidate specific interpretations. This is a difficult task for WSD processes, as the Semantic Web applies the open world assumption. Information not present might be correct and relevant. At the same time, the absence of information does not imply the incorrectness or meaningless-

ness of this information for the correct interpretation. For the purpose of WSD in the field of the Semantic Web, a negative context denotes irrelevant information for an interpretation. Thus, information pertaining to a negative context provides topics or relationships that rule out possible interpretations. In the case of the term *Berlin* from the example above, the negative context would contain the following topics respective of categories: *populated places in Colquitt County*, *towns in Georgia*, *towns in Hartford County*, *American songwriters*. For the correct interpretation of the term *Berlin*, semantic entities that pertain to the topics listed in the negative context are not considered. Therefore, the entities *Berlin in Georgia*, *Berlin in Connecticut* and *Irving Berlin, the songwriter* can be ruled out, because they pertain to relationships in the negative context. This example already shows that the construction of a negative context requires more semantic information compared to a positive context. While positive context items can comprise simple text terms, negative context items make use of semantic information, such as semantic entities, categories, or topics. The most simple way to provide a negative context is via manual composition of a blacklist (see Section 6.3.1) of prohibited semantic entities or categories for a specific application. This list of prohibited categories is used to penalize interpretations that are assigned to categories in the list or have other relationships to the negative context. The disambiguation is therefore performed in a subtractive manner. The more relationships an interpretation has to the negative context, the lower its relevance. The creation of a blacklist utilized as negative context requires considerable manual effort – especially for knowledge bases containing many entities and categories. Therefore, an automatic enrichment of the negative context enables a subtractive disambiguation process and is customizable for any context and knowledge base. The proposed approach of dynamic and automatic creation of a negative context and its utilization is described in Section 9.2. The evaluation of the negative context approach is presented in Section 10.7.7.

6.5 HETEROGENOUS CONTEXT

The extent of a context is constrained by its boundaries. In terms of multimedia documents, such as video, the context boundaries are defined by temporal divisions respective of fractions or segments of coherent content. Accordingly, all information pertaining to the same segment also pertains to a coherent context. This information can be manifold and of various data provenance. The context terms included in such a context possess various characteristics and must be handled correspondingly. Thus, the contexts created from video metadata are considered heterogenous contexts. On the contrary, a text document provides a homogenous context. All context terms occurring in the text document possess the same characteristics.

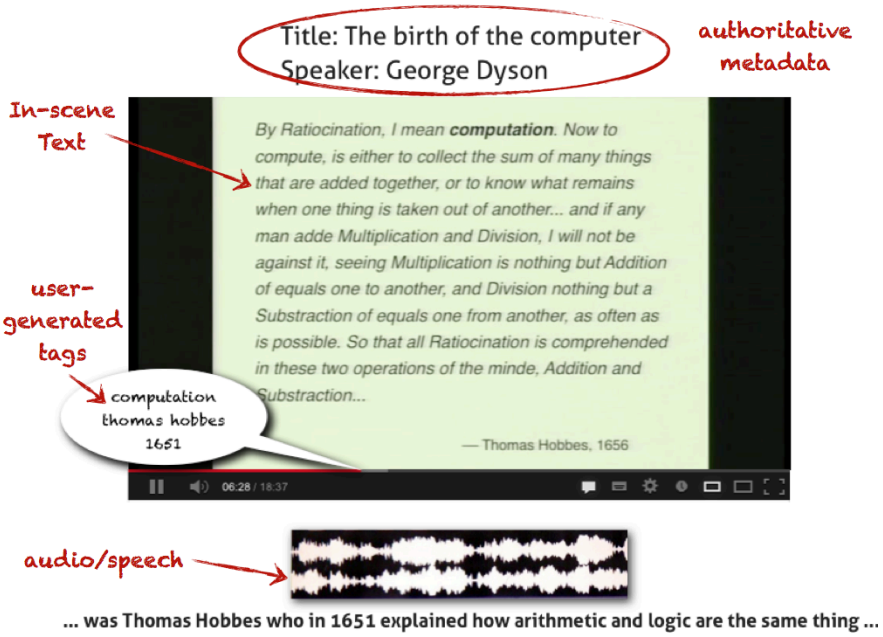


Figure 10: Example of video possessing metadata from different sources creating a heterogenous context

Figure 10 depicts a video possessing metadata from displayed and spoken text, user-generated tags and authoritative descriptive information. All textual information occurring at the same time or within the same structural segment pertain to the same context, but possess different characteristics. To describe these characteristics and utilize them for the subsequent semantic analysis processes, a context model for heterogenous contexts has been developed. The context model defines several characteristics to evaluate the context terms of a heterogenous context and bring them into an order with respect to their reliability and confidence. The context model is described in the next chapter.

7

CONTEXT MODEL FOR HETEROGENOUS CONTEXTS

The context model has been developed to handle different characteristics of context items in heterogenous contexts [107]. Video metadata is often of diverse data provenance and entails variant reliabilities. This chapter introduces the context model including descriptions of context items and deduced confidence values in Section 7.1. Contextual descriptions and confidence values are defined along examples and calculated using a sample of video metadata in Section 7.2. The deduced characteristics of the contextual descriptions enable multiple views on the context items. These views are introduced and described in Section 7.3.

7.1 CONTEXTUAL DESCRIPTION AND CONFIDENCE CALCULATION

Documents are created within a specific user context determining the purpose for which the document was created. This context can also be considered as pragmatics. The metadata provided for the document, as well as automatically generated metadata, form a separate context. This context is important for the determination of the interpretation of the information provided in the document. Therefore, context is defined as following:

Definition 7.1.1. A **context** is represented by a finite set CI of context items. Each **context item** $ci_i \in CI$ is a tuple $ci_i = (term, uri, cd, c)$, where:

- $term$ denotes the value (string text) of the context item ci,
- $uri = (uri_1, uri_2, ..., uri_n), uri_i \in KB, i = [1...n]$, denotes the list of n entity candidates assigned to the $term$ of the context item ci, where a semantic entity uri_i is part of a given knowledge base and is referenced by its URI (Uniform Resource Identifier) ,
- cd denotes the contextual description $cd \in CD$ of the context item ci, and
- $c \in [0...1]$ denotes the confidence value that is calculated according to the contextual description cd.

Thereby, a context is defined as a list of context items. The context items derive from a document's provided metadata or from textual content extracted from the document. Document metadata ranges

from structured data to continuous text. Type information provided with structured data is an important factor for the interpretation of the textual information. Most of the document metadata (both manually provided authoritative metadata as well as automatically extracted metadata) is presented in the form of natural language text. Natural language is expressive but entails the problem of ambiguity. To enable semantic annotation of documents and the documents' metadata, the ambiguity of the textual information must be resolved. This is where the context comes into play. The characteristics and ability for interpretation of a context are determined by the sum of the context items pertaining to it. Importantly, these context items originate from diverse sources, can have different reliabilities and should therefore be weighted according to their significance within a context. This significance is represented by the calculated confidence value c associated with the context item ci_i. In order to achieve this, a contextual description depicting the characteristics of context items is defined:

Definition 7.1.2. A **contextual description** $cd \in CD$ of a context item ci is a tuple $cd = (tt, st, sd, cl, nt)$, where:

- CD denotes the set of all contextual descriptions,
- $tt \in Tt$, where Tt is a finite set of text types, and tt is associated with the context item ci,
- $st \in St$, where St is a finite set of source types, and st is associated with the context item ci,
- $sd \subseteq Sd$, where Sd is a finite set of available sources for the document,
- $cl \in Cl$, where Cl is a finite set of ontology classes, cl is the class the entities in uri of context item ci are type of, and
- nt denotes the number of tokens of the $term$ of associated context item ci.

For the proposed use case, the semantic analysis of video metadata, text types, sources, and ontology classes of the contextual descriptions are restricted to the following sets:

- The set of source types St is determined to contain authoritative and non-authoritative (human) sources, as well as Automatic Speech Recognition (ASR), and Optical Character Recognition (OCR).
- The set of text types Tt is determined to contain continuous text, keywords, and tags.
- The set of ontology classes Cl is determined to contain place, organization, and person.

Sources of automatically extracted textual information from video data include displayed and spoken text and therefore the respective OCR and ASR algorithms (see Section 2.4.2 and 2.4.3). Often, minimal authoritative metadata is available, such as a title, speaker, primary persons, publisher, etc. Additionally, some video resources are provided with textual, time-related tags from non-authoritative sources[1]. Therefore the set of available source types for video metadata $\mathtt{St}$ is restricted to these sources: authoritative, non-authoritative, OCR and ASR.

Metadata from ASR and OCR algorithms, as well as the title and description from the authoritative metadata, can be considered as continuous text, i.e. textual data without type information that requires further linguistic analysis for interpretation. Information about the speaker or the publisher of a video are typically given as keywords, i.e. a term or a list of terms, where each term corresponds to a semantic entity. Tags form a third text type as they are mostly given as a group of single terms and only subsets of the group belong together (see Section 8.2.1 for tag processing). Interpretation of tags also requires further linguistic or statistical analysis. $\mathtt{Tt}$ is therefore restricted to these three text types: continuous text, keywords, and tags.

To determine the appropriate entities for a given textual information, it is advantageous to know each entitiy's prospective ontology class. Some authoritative metadata can be directly assigned to ontology classes, such as the metadata item for *speaker* can be directly assigned to the ontology class *Person*. For NLP, Conditional Random Field (CRF) classifiers (see Section 3.2.2) are applied to find entities of predefined ontology classes in continuous text. By using a 3-class model, the ontology classes *Person*, *Place* and *Organization* can be recognized in a text. Therefore, in this use case the set $\mathtt{Cl}$ is restricted to these three ontology classes.

With the help of the contextual description, the confidence of the context item can be computed. For each of the five contextual factors ($\mathtt{tt}, \mathtt{st}, \mathtt{sd}, \mathtt{cl}$, and $\mathtt{nt}$) a double precision value is calculated within a range [0.0...1.0].

The five factors can be aggregated to superordinate characteristics of two video metadata aspects: correctness and ambiguity.

7.1.1 *Correctness*

By the correctness of a context item, the probability that the context item is correct with respect to syntax and relevance is denoted. The correctness of a context item is influenced by the source diversity as well as by the source reliability. The more sources agree on a textual

1 video portals like Yovisto allow the videos to be tagged by any user to provide time-related references to the video

information[2] and therefore an item, and the higher the reliability of the item's original source, the higher the reliability that this item is correct. The two contextual factors source reliability and source diversity are described in detail in the following paragraphs.

SOURCE RELIABILITY The term *reliability* refers to a prospective error rate concerning the source type st. Document metadata can be created by either human or computer agents. Human agents can be the author who created the document, or any user who annotated the document with additional information. Computer agents are analysis algorithms, which extract (mostly) textual information from a multimedia document, such as OCR and ASR.

All these agents provide information with differing degrees of reliability. Where human agents in general can be considered more reliable than computer agents because of linguistic knowledge and experience, authoritative human agents are considered more reliable than non-authoritative human agents. An agent's reliability is ranked according to this simple presumption.

The value ν_{st} denotes the probability of accuracy of a context item with respect to its source type st. We assign the highest possible value $\nu_{st} = 1.0$ to authoritative human sources as being the most reliable source type. Non-authoritative human sources are considered less reliable than authoritative sources [105]. Therefore, a 10% penalty is assumed and $\nu_{st} = 0.9$ is assigned. Please note that these values only serve as an assumption and can be endorsed by empirically generated trust values for specific users or user groups. For computer agents the reliability values from determined accuracies respective of the quality of the results of the algorithms are adopted. Unfortunately, most video OCR evaluations are based on single frame processing, which embellishes the achieved results. Precision for video OCR on videos with an equal number of text and non-text frames is still very low. According to [119], the error rate for news videos can be as high as 65%. Therefore, a worst case accuracy of 35% ($\nu_{st} = 0.35$) for context items with an OCR analysis as source agent is assumed. Word error rates for ASR analysis engines range between 10% and 50% (accuracy rates between 50% and 90%)[24]. The worst case of 50% accuracy is assumed and the reliability value for context items from ASR results to $\nu_{st} = 0.5$ is determined. The assigned reliability values for the different source types are depicted in Table 3.

SOURCE DIVERSITY Source diversity specifies how many of the available annotation sources agree on the same metadata item. Thus, the same textual information is provided by different sources. The diversity ranges from a single source to all available sources. The

2 That means, multiple sources provide the same textual term.

Table 3: Overview of assigned reliability values for different source types of a context item

source type	v_{st}
authoritative human source	1.0
non-authoritative human source	0.9
ASR	0.5
OCR	0.35

more sources agree on the textual term of a context item, the more reliable the item is considered to be.

Depending on the number of available sources (Sd) and the set of sources that agree on the same item i (s_i), the value v_{sd} denoting the probability of accuracy of a context item ci with respect to its source diversity can be calculated as follows:

$$v_{sd} = \frac{|s_i|}{|Sd|} \tag{1}$$

Example: The text *operating systems* is automatically extracted by OCR analysis from a video frame. The title of the video is *Operating Systems: Lion vs. Jaguar*. For this video, the only sources of textual information are the authoritative metadata and the extracted OCR text. For the authoritative metadata, a context item with $term$ =*operating systems* can be derived, while this context item is also supported by OCR. In this case as the term *operating systems* is confirmed by both available sources, v_{sd} calculates to $v_{sd} = \frac{2}{2} = 1.0$.

7.1.2 *Ambiguity*

The ambiguity level of a context item is influenced by the text type, the ontology type assigned to the item, and the number of tokens the context item's natural language term comprises. The level of ambiguity is also relevant during successful disambiguation in terms of a correct interpretation. Therefore, the lower the ambiguity, the higher the possibility of successful interpretation of a context item.

Natural language text needs NLP technologies to identify important key terms. Due to the number of potential errors, the ambiguity of natural language text is often higher than for single restricted key terms. For key terms, no further processing is needed. Also, the lower the amount of instances of the assigned ontology class, the lower the item's potential ambiguity. The number of tokens the term consists of gives a hint about the specificity of the term. The more tokens, the more specific the term might be.

Table 4: Overview of assigned confidence values for different text types of a context item

text type	v_{tt}
key terms	0.9
tags	0.7
continuous text	0.56

TEXT TYPE Video metadata is provided in different text types. In general, descriptive texts as well as automatically extracted texts are continuous texts. Authoritative metadata can also be provided as typed key-value pairs, or untyped keywords. These key terms usually depict an entity in total. Further authoritative textual information, such as the title or a descriptive text, are given as continuous text in natural language. It is assumed that the ambiguity of metadata items with the text type "typed literal" is lowest, because no detection algorithm is needed and the type of the literal is already given. Therefore, the according confidence value v_{tt} denoting the probability of ambiguity of a context item ci with respect to its text type is highest with $v_{tt} = 1.0$. In reality, this text type is usually not representative for video metadata and therefore are not part of the set of available text types. The ambiguity of continuous text depends on the precision of the NLP algorithm used to extract key terms. For this work, the Stanford POS tagger[3] is applied to identify word types in text. This tagger has an accuracy rate of 56% per sentence [66], which leads to $v_{tt} = 0.56$. By using this rate as a reliability value for continuous text, a measure independent of text length is applied. POS tagging is not needed for context items that are provided as key-value pairs. Still, to allow for uncertainty the reliability of key terms is determined to be slightly lower than for typed literals as $v_{tt} = 0.9$. Key terms therefore achieve the highest v_{tt} within the scope of video metadata. User-generated tags require further linguistic and statistical processing to find named entities in tag groups (see [63] for tag processing). The precision of detecting key terms in tag groups has been determined empirically as 0.7. Therefore, the confidence value for tags is set to $v_{tt} = 0.7$. The assigned confidence values for different text types are depicted in Table 4.

CLASS CARDINALITY The contextual factor of class cardinality corresponds to the number of instances the assigned ontology class contains for the context item ci. In general, a descriptive text does not refer to a specific ontology class if a CRF classifier does not find any classes in the text. The entities found in such a text can be of any type. In such a case, the context items found in this natural language

3 http://nlp.stanford.edu/software/tagger.shtml

text are assigned to the most general class $\top$ of the ontology[4] and the class cardinality is highest.

According to the context item's assigned ontology class cl and its known cardinality the value v_{cl}, denoting the probability of ambiguity of a context item ci with respect to class cardinality is calculated proportional to the overall number of all known entities ($|\top|$). $\top$ denotes the most general class containing all individuals of the knowledge base, and $|\top|$ denotes the number of all instances pertaining to this class. A high class cardinality entails a high ambiguity. Therefore, the value v_{cl} is inverted to reflect a reverse proportionality regarding the amount of the value and the ambiguity:

$$v_{cl} = 1 - \frac{|cl|}{|\top|} \tag{2}$$

Within the scope of video metadata only three different ontology classes are applied: person, place, and organization. The fourth v_{cl} relevant for video metadata is when no ontology class is assigned. Typically, the class cardinalities for the three pertinent ontology classes differ only slightly. For the purposes of simplicity v_{cl} is calculated in linear proportion to the cardinality of the assigned class. To demonstrate, a context item of a video might be identified as Person (by its author or by an automatic NER tagger). Using the DBpedia Version 3.8.0 as a knowledge base, the class *Person* contains 763,644 instances. The top class of the DBpedia ontology, `owl:Thing`, holds 2,350,907 instances. Accordingly, the confidence value $v_{cl} = 1 - \frac{763,644}{2,350,907} = 0.67$ for a context item assigned to the DBpedia ontology class *Person*.

The number of entity candidates of a term might also be a measure for the prospective ambiguity of the term. However, evaluations showed better results for the approach on class cardinality. Details on the evaluation results are described in Section 10.7.5.

NUMBER OF TOKENS Labels of semantic entities often consist of more than a single token. The more tokens an extracted natural language term comprises the more specific this term might be considered. Therefore, the prospective ambiguity of a term declines with the increasing number of its tokens. Confidence values are retrieved through the frequency of terms with the respective number of tokens in the underlying dictionary for entity look-up[5]. The confidence value v_{nt} is calculated as follows:

$$v_{nt} = 1 - \frac{c_{nt}}{c_{all}} \tag{3}$$

4 which means, all entities of the knowledge base have to be considered and the amount cannot be restricted to a specific class

5 Some approaches to entity look-up dictionaries are presented in Section 8.1.1 and various dictionaries are introduced in Section 10.5.

Table 5: Confidence values for different numbers of tokens according to their frequencies in the dictionaries *SPL* and *GCW*. Details of the dictionaries are presented in Section 10.5.

Number of Tokens	v_{nt}^{SPL}	v_{nt}^{GCW}
1	0.57	0.67
2	0.63	0.83
3	0.87	0.89
4	0.95	0.92
5	0.98	0.95

c_{all} denotes the distinct number of all terms in the dictionary and c_{nt} denotes the number of terms possessing the respective number of tokens. Table 5 shows sample values for different numbers of tokens according to two sample dictionaries.

The confidence values of the five constituents of the contextual description can be computed as a linear combination and normalized to a range of [0.0...1.0]:

$$c = \frac{v_{st} + v_{sd} + v_{tt} + v_{cl} + v_{nt}}{5} \tag{4}$$

For reasons of simplicity, the average of equally weighted constituents has been chosen. Section 10.7.5 discusses several surveillances for emphasizing individual constituents compared to the other constituents.

7.2 CONFIDENCE CALCULATION FOR CONTEXT ITEMS

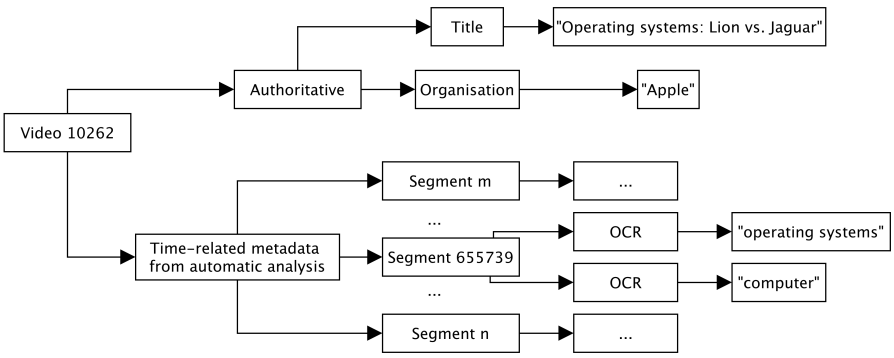


Figure 11: Example of various types of metadata for a video document

Let an example video have the following authoritative metadata information:

- Title: *Operating systems: Lion vs. Jaguar.*
- Publisher: *Apple*

Table 6: Example values for contextual factors and the according confidence for the six context items of the example

term	tt	v_{tt}	cl	v_{cl}	st	v_{st}	v_{sd}	v_{nt}	c
Apple	keyword	0.9	Organization	0.96	authoritative	1.0	0.5	0.0	0.652
Operating systems	continous text	0.56	$\top$	0.0	authoritative	1.0	1.0	0.33	0.558
operating systems	continous text	0.56	$\top$	0.0	OCR	0.35	1.0	0.33	0.448
Lion	continous text	0.56	$\top$	0.0	authoritative	1.0	0.5	0.0	0.392
Jaguar	continous text	0.56	$\top$	0.0	authoritative	1.0	0.5	0.0	0.392
computer	continous text	0.56	$\top$	0.0	OCR	0.35	0.5	0.0	0.282

Additionally, *computer* and *operating systems* were extracted from the video via OCR analysis. This example is depicted in Figure 11.

Publisher information is considered as a keyword. Publisher is assigned to the DBpedia ontology class *Organization*. The title and the OCR texts are considered as continuous text. The NER tagger did not find any class types in the title or the OCR information. After NLP pre-processing, six context items are generated from the given metadata. The contextual factors and the calculated confidence values of the six context items are shown in Table 6.

7.3 CONTEXT ITEM VIEWS

As shown in Figure 12, the identified contextual factors and dimensions influence different superordinate characteristics and can be aggregated in two different views: the confidence view and the relevance view. The confidence view aggregates the characteristics of a context item described above. It should be noted that context items in heterogenous contexts also have characteristics regarding their context relevance.

7.3.1 *Confidence View*

The confidence of a context item is influenced by its ambiguity and correctness, as described in Section 7.1. With the term confidence the trust level assigned to the item for further analysis steps is denoted. A high correctness and a low ambiguity entail a high confidence for the context item. The confidence view is applied to order context items according to their accuracy and ambiguity. The higher the confidence the higher is the probability that the context item is analyzed correctly and the correct entity is assigned to the item.

7.3.2 *Relevance View*

The various context boundaries described in Section 6.2 span different dimensions for context items and determine how relevant they are for the analysis of other context items. The spatial, temporal and structural dimensions specify the relevance of a context item in relation to other context items. These dimensions help to identify the divergence of the context items with respect to the document's content. When creating a context for a semantic analysis of context items, the relevance view is important to aggregate the amount of all context items related to a document into smaller groups of stronger content-related coherence. This allows for a more accurately and therefore more meaningful semantic analysis of context items.

Metadata items of time referenced documents such as video or audio files can be assigned either to document fragments or to the full

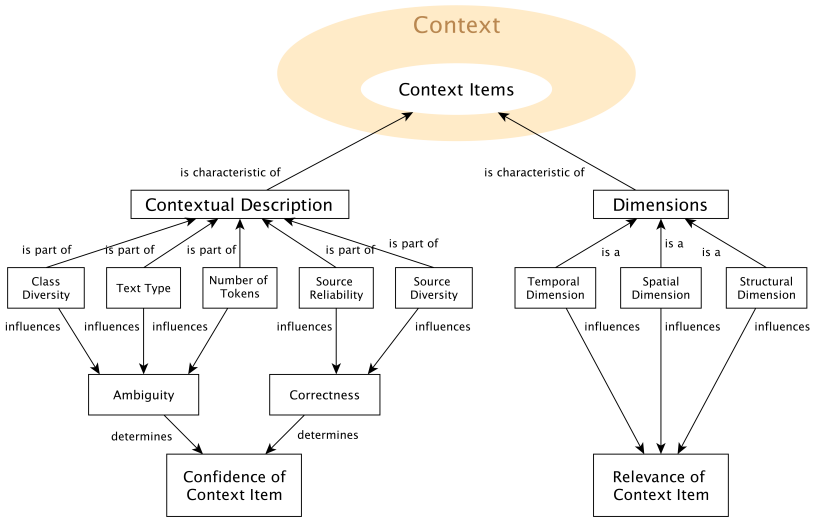


Figure 12: Contextual factors of context items

document. The **temporal dimension** reflects the reference period of the metadata item. The values of this dimension have a range between the smallest unit of the document (e. g., a frame for a video) and the full document. The **spatial dimension** assigns the metadata item either to a specific region or to the entire document. That is, for a video document the values begin with a single pixel within a frame over a "geometrically determined region within a frame" to the full frame. The **social dimension** plays a special role within the characteristics of context items. It takes into account information about the social relationships of the user who created the metadata item, as well as the user who accesses the document (or even a person referenced in a document). Therefore, the social dimension relates to the user and covers a personal perspective.

For the proposed semantic analysis process of video metadata, the relevance view refers to the temporal dimension of the video. Context items referring to the same content-based segment are considered to belong to the same context. Those context items possessing the same reference to a video segment are aggregated into a heterogenous context. The *reference* in Table 10 (see Section 9.1.1) shows the context items of the video metadata example and the temporal reference with respect to the relevance view.

8

SEMANTIC ANALYSIS APPROACH IN GENERAL

This chapter introduces the proposed semantic analysis process in general. The description is aligned with the four elements of a WSD process (see Section 3.2.3). In this chapter three of these four elements are introduced: selection of word senses, application of external knowledge sources, and the actual disambiguation method. The first two elements are discussed in Section 8.1; the third element is described in Section 8.2. The fourth element – representation of context – is addressed in Chapter 9 when the semantic analysis process is combined with the proposed context model.

Section 8.1 gives an overview of the knowledge base that is required for a general WSD process within the scope of the semantic web. The knowledge base must provide a dictionary to look up entity candidates for natural language terms of the input text. Additionally, disambiguation data – descriptive data about the entities – is required to perform a disambiguation, if needed. Ambiguous natural language terms must be disambiguated to identity the correct meaning of the terms and assign the most fitting entity for the present context. For the actual disambiguation process, several analysis methods have been developed. These methods are introduced in Section 8.2.

8.1 CONSTRUCTION OF A KNOWLEDGE BASE

The basis of the WSD algorithm is the dictionary containing word senses (entities and their textual representations) and the underlying knowledge base containing descriptive information about the entities.

Ideally, the dictionary consists of complete sets of textual representations for every known entity. During the WSD process the dictionary serves as a reverse look-up index to find entity candidates for a term detected in the text. Several approaches for the set-up of a dictionary are presented in Section 8.1.1. Section 8.1.2 describes a scoring algorithm to rank all textual representations of a semantic entity regarding their objective, context-independent relevance for the entity.

The knowledge base includes descriptive information about the entities used to distinguish different entity candidates during the disambiguation process. The extraction of this disambiguation data is described in Section 8.1.3.

8.1.1 *From Term to Entity - Finding Labels*

Within the scope of the Semantic Web, an entity is referenced by its URI and named by a main label. However, in natural language texts, entities are often mentioned by synonyms, acronyms, abbreviations of the main label or other alternative idioms. Some Linked Data datasets, such as the GND, provide alternative labels in addition to main labels. For other datasets, such as DBpedia, multiple sources are utilized to compose a preferred complete set of textual representations for each entity. The textual representations include main and alternative labels of the entities. The set of textual representations constitutes the dictionary that can be looked up for possible entity candidates. In the following paragraphs, two approaches to the construction of an entity dictionary from DBpedia information are presented.

DISAMBIGUATION LINKS AND REDIRECTS DBpedia provides information about redirects and disambiguation pages of all entities. Disambiguation pages in Wikipedia are summaries of articles with the same main label and list links to the respective articles. Articles with ambiguous main labels often possess a title containing disambiguating information, such as *Emma (1996 TV drama)*, or *Emma (2009 TV serial)*. Very often, such disambiguation pages list people with identical surnames or places with the same name in different countries or regions. Therefore, the label of the disambiguation page often constitutes the main label without the brackets used for distinction of entities with the same name; or the page represents the abbreviation that stands for several different entities. The following disambiguation pages all point at the entity `dbp:Brooklyn_Nets`:

- `dbp:NET`
- `dbp:NJN`

The labels *NET* and *NJN* are added as alternative labels accordingly.

Additionally, the labels of redirects represent an essential source for alternative names of entities. Redirects evolve in different ways:

- A user creates a page for an entity that already exists, but with a different title.
- The article title is changed.
- The redirect is created manually to lead the user to the correct article when searching for an alternative name of the entity.

Redirects possess a unique label but no other information about the target entity. DBpedia 3.8.0 lists 9 redirects for the entity `dbp:Brooklyn_Nets`, e. g.:

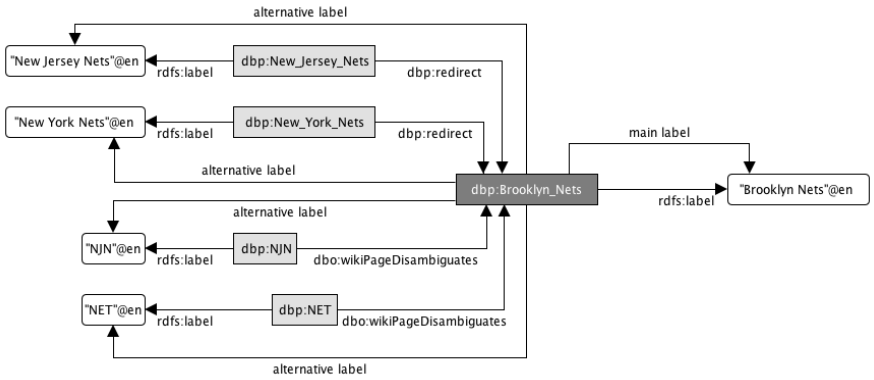


Figure 13: Retrieval of alternative textual representations for DBpedia entities using redirects and disambiguation links.

- `dbp:New_York_Nets`,
- `dbp:New_Jersey_Nets`, and
- `dbp:New_Jersey_Americans`.

Accordingly, the labels *New York Nets*, *New Jersey Nets*, and *New Jersey Americans* are extracted as alternative textual representations.

Figure 13 shows the retrieval of alternative labels for DBpedia entities. Table 7 shows the list of extracted alternative textual representations for the entity `dbp:Brooklyn_Nets` using redirects and disambiguation pages (the presented scores for the labels are introduced in Section 8.1.2). In addition to the labels for redirects and disambiguation pages, for persons supplementary labels can be extracted. Persons are often only referred to by their family names. Therefore, they can be extracted from the main labels as an additional label. For example, for *Steve McQueen* his family name *McQueen* is extracted and added as an alternative person name. References to aristocratic persons form a special case. They are often only referred to by their given names. For example, for *Catherine, Duchess of Cambridge* only *Catherine* is extracted as an alternative person name.

EXTRACTION OF ANCHOR TEXTS The second approach is based on the assumption that alternative labels of entities can be extracted from the anchor texts[1] when the entities are mentioned and linked in natural language texts. The success of Wikipedia is mainly based on the fact that text articles contain links to other articles and the encyclopedia can be explored by simply following these links. Therefore, authors of Wikipedia articles are urged to provide links to other articles when mentioned in the article texts. The anchor texts of these links can be extracted and utilized as dictionary entries for textual representations of the entities. Table 8 shows a list of mentions of the

1 The text used as an anchor to provide a link to another article.

Table 7: Textual representations of `dbp:Brooklyn_Nets` using labels of redirects and disambiguation pages and the calculated scores.

Label	Label Score
Brooklyn Nets	0.33
New York Nets	0.66
New Jersey Nets	1.0
New York Freighters	0.42
New Jersey Americans	0.65
Nj nets	0.466
NJ Nets	0.466
Sly Fox	0.125
NJNets	0.399
NJN	0.8

entity `dbp:Brooklyn_Nets` in the articles of the English Wikipedia[2] (the presented $p(article|anchor)$ is introduced in Section 8.1.2).

DISCUSSION Anchor text extraction provides many more alternative labels for an entity than provided by redirects and disambiguation links. However, persons are mostly referenced by their full name. The Wikipedia policies recommend that users embed a link to another article at its first appearance within the current article. Therefore, when persons are linked, the anchor text often contains their full name. A manual extraction of alternative person names (as described in *Disambiguation Links And Redirects*) provides additional labels. The main labels of entities of the type *Person* might be analyzed. The family name of the person might by used as an alternative label in addition to the main label containing the full name. However, the analysis for family names is difficult and often fails for aristocratic persons (such as *Prince of Wales*) or persons with a pseudonym (such as *Lady Gaga*). Also, the anchor text approach involves much effort, because the underlying text corpus has to be parsed regularly to extract anchor texts in and for newly created articles.

Section 10.5 introduces four different dictionaries created by the two presented approaches. Evaluation regarding the characteristics, coverage and size of the dictionaries have been processed and are presented.

8.1.2 *Label Ranking*

As shown in the previous section, entities can be identified by multiple labels and labels can likewise identify multiple entities. Thereby,

2 As of October 2011.

Table 8: Mentions of the entity `dbp:Brooklyn_Nets` in the articles of the English Wikipedia and the respective probability $p(article|anchor)$

Label	$p(article\|anchor)$
New York	0.00004
New Jersey Nets Summer League Team	1.0
NY Nets	1.0
New York/New Jersey Nets	1.0
New York / New Jersey Nets	1.0
NETS Basketball	1.0
Brooklyn Nets	1.0
NJ	0.00073
New Jersey Americans/New York Nets	1.0
New Jersey Nets	0.98933
Nets	0.38994
New York Nets	0.99367
Brooklyn Sports & Entertainment	1.0
N.Y. Nets	1.0
New York Americans	0.00076
NJN	0.45455
NYN	0.25
New Jersey	0.01285
New Jersey Americans	0.61702
New York Nets/New Jersey Nets	1.0

some labels might be more relevant than others for an entity and some entities might be more or less relevant for a label. This fact can be represented by a label ranking. In the following discussion, two approaches to label ranking are described – based on the two approaches to dictionary construction presented in the previous section.

ANCHOR-TEXT-RANKING The Wikipedia article corpus including the *interwiki*[3] links provides a source of the probability of a linked article (entity) for a given anchor text. As shown in Table 8, an article can be referenced by multiple anchor texts. The probability of a linked article for a given anchor text $p(article|anchor)$ is calculated as follows:

$$p(article|anchor) = \frac{p(article \cap anchor)}{p(anchor)} \tag{5}$$

where $p(article \cap anchor)$ refers to the probability that $article$ is linked using $anchor$ and $p(anchor)$ is the probability for $anchor$ being an anchor text for any link. The probability of an anchor text for a given linked article is calculated respectively:

$$p(anchor|article) = \frac{p(article) \cap anchor}{p(article)} \tag{6}$$

For NLP – and WSD in particular – the challenge is to find the best fitting entity for a given natural language term. Therefore, the probability presented in Equation 5 calculates the more relevant score for this purpose compared to the probability presented in Equation 6, because it represents the probability that a specific article is meant for a given text. As an example, the probabilities $p(article|anchor)$ for the anchor texts linking to the article for the entity `dbp:Brooklyn_Nets` are presented in Table 8.

RANKING OF ALTERNATIVE LABELS The anchor text ranking described above cannot be applied for the alternative labels derived from redirects and disambiguation pages or given as alternative labels, because some of these alternative labels might not be used as anchor texts to point to articles or a respective text corpus may not be available. Therefore, the calculation of the probability p of an article for a given alternative label is not applicable.

As an alternative, the label ranking approach is based on the assumption that the main label is the best fitting label for an entity. Derived alternative labels are ranked according to several rules comparing main and alternative labels. Every label achieves a score within the range [0.0 ... 1.0].

3 Internal links between the articles within the Wikipedia. Links to external resources are *not* considered.

In a preprocessing step, all WordNet *synsets*[4] and acronym lists are retrieved. For the label ranking calculation, the following queries for every pair of main label and alternative label are executed in the presented order. If a query is true, the respective score is assigned to the alternative label and the subsequent queries are skipped:

- Does the main label equal the alternative label?
  $score = 1.0$
- Are both the main label and the alternative label in the same WordNet synset list?
  $score = 0.9$
- Does the main label refer to a person and is the alternative label an alternative person name[5]?
  $score = 0.9$
- Is the alternative label an acronym of the main label?
  $score = 0.8$
- Does the main label contain the alternative label?

  $$score = \frac{length(alternativelabel)}{length(mainlabel)}$$

The main label is considered the best fitting label and achieves the highest $score = 1.0$. A synonym of the main label achieves a slightly lower score as the main label. Alternative person names (such as the family name) are treated as synonyms of the full name and achieve the same score. Acronyms are considered more ambiguous and less relevant compared to synonyms. Under this assumption, the score for acronyms is defined to $score = 0.8$. If the alternative label is not a synonym or an acronym, but a substring of the main label, the proportional length of the alternative label to the length of the main label is used as a score. If none of the previous queries returned true for the pair of main and alternative label, the score is calculated in the following way:

$$score = 1 - \frac{levenshtein(mainlabel, alternativelabel)}{max(mainlabel, alternativelabel)} \tag{7}$$

where $max(mainlabel, alternativelabel)$ returns the length of the longer of both labels and $levenshtein(mainlabel, alternativelabel)$ returns the Levenshtein distance[6] for the main and the alternative label. As an example, the scores for the alternative labels of the entity `dbp:Brooklyn_Nets` are presented in Table 7.

4 A WordNet synset is a list of textual terms representing the same entity.
5 see Section 8.1.1 for extraction of alternative person names.
6 Levenshtein distance of two labels refers to the minimum required number of edits to convert one label into the other. An edit is an insertion, deletion, or substitution of a single character.

DISCUSSION Label ranking provides a context-independent measure for ranking entity candidates for a given natural language term. The anchor text approach is only applicable if alternative labels are derived by using anchor texts of links in a large text corpora. This label ranking reflects the manner of linking in the text corpora. Within the scope of Wikipedia, several constraints and characteristics have to be considered. On the one hand, Wikipedia itself imposes several rules for the creation of articles. Therefore, the users cannot choose freely where and what to link. On the other hand, the users creating Wikipedia articles cannot be considered Semantic Web experts and they do not create the articles considering the effect of their work on extraction methods. Therefore, an automatic extraction of labels and the applied anchor ranking might comprise noise and incorrect data. In contrast, the latter label ranking approach does not consider common use of labels for entities (as does the anchor text ranking). This might result in higher scores for labels that are commonly used in (web) documents to refer to the specific entity. Likewise, more popular labels (with respect to their usage as anchor texts) might receive lower scores, because of missing synonym information and a low lexical similarity. Section 10.6 provides several statistics for entity dictionaries and their respective label rankings applied on different benchmarks.

8.1.3 *Disambiguation Data*

The definition of an entity and its distinction from other entities with similar names is represented by descriptive metadata. This metadata can be used for the disambiguation process. Descriptive metadata is on the one hand natural language text describing the purpose, facets and history of the entity and on the other hand relationships to other entities. The composition of descriptive texts as well as different types of entity relationships are presented in the next paragraphs.

CO-OCCURRENCE OF ENTITIES AND TEXT Descriptive natural language texts can be derived in various ways. Articles from Wikipedia or other sources provide natural language texts describing a given entity. In comparison, an entity can also be described by the texts with which it co-occurs. Therefore, the text parts surrounding a link to an entity can also be considered as descriptive texts. Hereby, the context size has to be defined. The surrounding texts can be limited by sentence, paragraph, section or document boundaries. The larger the context, the more irrelevant information might be considered, in terms of describing a linked entity. A context which is too small might provide too little information for a sufficient disambiguation and therefore correct interpretation.

ENTITY RELATIONSHIPS Within Wikipedia, related articles are represented by the page link graph. The graph consists of nodes representing articles, and links between articles represent the edges. If an article refers to another article, the respective nodes in the graph are connected by a directed edge. This graph contains all links between articles extracted from the article texts. However, the links are not further defined regarding the type of relationship of the two involved nodes.

In contrast, the property graph (in this case extracted for the DBpedia from the infoboxes) only represents a small subset of this very large link graph. The property graph constitutes the entities that are linked using an ontology property extracted from the articles' infoboxes. Thereby, the type of relationship is determined by the used property.

Although the page link graph provides untyped relationships between entities, it reveals more information about an entity and the entities it is related to. For instance, taking into account only the property graph, `dbp:Catherine,_Duchess_of_Cambridge` is related to `dbp:Reading,_Berkshire`, `dbp:Prince_William,_Duke_of_Cambridge`, and `dbp:House_of_Windsor`. At the same time, the Duchess is linked to 452 entities via a page links. For example, there is a relation to `dbp:University_of_St._Andrews`, because she graduated that institution, and there is a connection to `dbp:Alexander_McQueen`, because she once wore a dress created by the designer.

In summary, entity relationships represented by page links provide essential knowledge about an entity, beyond the knowledge provided by extracted triples via info box properties. However, such extensive relationship graphs can only be derived from crowd-sourced text corpora, such as Wikipedia. In contrast, the GND (see Section 4.1.4) only contains typed relationships, such as related places, persons or superordinate concepts. The GND lacks a large text corpus containing links to other entities and extensive descriptive texts. For that reason, only the typed relationships can be used as distinctive disambiguation data.

8.2 DISAMBIGUATION OF NAMED ENTITIES AND COMMON WORDS

The process introduced subsequently corresponds to WSD as described in Section 3.2.3, but within the scope of the Semantic Web and under consideration of Linked Data and semantic entities. For the purpose of the analysis of video metadata, the available textual information is analyzed for named entities as well as for other common words and combined terms that are important to describe the video's content. Thus, the following process takes NED a step further by also extracting and disambiguating crucial key terms. For the purpose of sim-

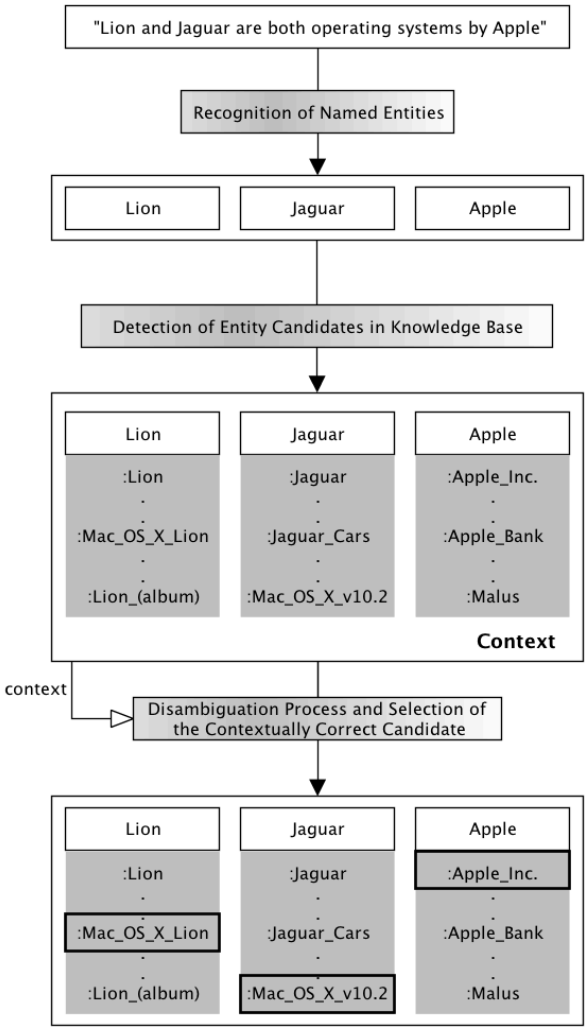


Figure 14: Overview of the disambiguation process

plicity, both named entities and other important key terms extracted from textual information are subsequently referred to as *entities*.

In general, the proposed context-based disambiguation algorithm comprises three different steps that build upon each other. The steps correspond to the process introduced for WSD in Section 3.2.3:

1. Identification of named entities and key terms
2. Detection of potential entity candidates in the underlying dictionary
3. Disambiguation and selection of the best matching candidate under consideration of the context

The first step identifies prospective entities and key terms in the given text. This process is described in Section 8.2.1. In the next step, entity candidates are retrieved in the given knowledge base for

each previously detected entity. The detection of entity candidates is shortly introduced in Section 8.2.2. Subsequently, entities that have multiple candidates assigned must be disambiguated by making use of the given context. For the disambiguation process, several main analysis methods have been developed:

- Co-Occurrence Analysis
- Link Graph Analysis
- Coreference Analysis

These analysis methods are described in Sections 8.2.3 through 8.2.5. Additionally, several context-independent methods can be used to detect the most fitting entity candidate. These additional methods are described in Section 8.2.6. All disambiguation methods return a disambiguation score within a range [0.0...1.0]. Some of the introduced methods might be more indicative than others. Therefore, applying a weight to the achieved scores is discussed in Section 8.2.7.

The workflow of the general process is depicted in Figure 14.

8.2.1 *Identification of Named Entities and Key Terms*

The three identified text types – continuous text, tags and keywords – require different methods to recognize named entities and key terms. Continuous text and user-generated tags require special attention for the identification. These two approaches are described in the next paragraphs.

The Oxford dictionaries define a keyword as *a word used in an information retrieval system to indicate the content of a document* and *a significant word mentioned in an index*[7]. Therefore, a keyword is considered representative of important key terms or named entities. Metadata marked as keyword therefore do not require any further preprocessing with respect to identification. A keyword is looked up directly in the underlying dictionary.

NATURAL LANGUAGE TEXT For the recognition of entities within continuous natural language text a lexical analysis is required (see Section 3.1). Existing libraries, such as *openNLP* or *Stanford NLP tools* process the text and assign every token to a specific word category. Named entities and important key terms are mainly nouns and combinations of nouns and other word categories. Therefore, a few combinations of word categories are relevant for the recognition of entities. The identified combinations and respective examples are listed in Table 9.

7 http://www.oxforddictionaries.com/definition/english/keyword

Table 9: Combinataions of word categories which are relevant for the recognition of key terms and named entities

Word Category Combination	Example
simple noun	motor
combined noun	motor oil
proper noun	Berlin
combined proper noun	New Jersey Nets
adjective followed by a noun	functional programming
(proper) nouns connected by a preposition	University of California

The identification step provides a list of natural language terms found in the input text. These terms match the identified word categories or combinations of word categories as shown in Table 9. In the next step, entity candidates for the terms are looked up in the underlying dictionary. This process is described in the next section.

USER-GENERATED TAGS User-generated tags are provided in multiple forms. Mostly, they are given as a group of single terms and only subsets of the group belong together (see Section 2.3). The proposed combination algorithm considers all tags of a specified temporal context and generates every possible combination of at most three terms within the context in any order. Reasons for this specific number are explained later on. Thus, it is assured that the algorithm combines groups of single terms that belong together. The number c of possible combinations is calculated as follows:

$$c = \sum_{k=1}^{j} \frac{n!}{(n-k)!} \tag{8}$$

About 90% of the DBpedia labels consist of at most three tokens, but less than 5% consist of 4 words. Due to these numbers and performance issues the number of terms to be combined has been limited to three. More results on dictionary analysis on the numbers of tokens for the labels are presented in Section 10.6.4.

As an example, the following tags have been annotated at the same timestamp for a video: *hubble*, *spitzer*, *carbon*, *dioxide*, *methane*, *co2* and *water*. For this sample context containing seven tags and at most three terms in a combination ($j = 3$), 259 combinations have to be generated. All combinations are looked up in the knowledge base and the combinations featuring entity candidates are maintained as terms.

ANNOTATION OF PARTIAL ENTITIES VERSUS COMPLETE ENTITIES Combined (proper) nouns represent an entity in the aggregate, but

their individual component nouns might also be assigned to entities separately. For instance, a text containing *John F. Kennedy Airport New York City* might be assigned to the entity for the airport in New York City, but *John F. Kennedy* might be assigned to the former president, and *New York City* to the city in the American state of New York. Since the document mentions the airport in New York City, and neither the former U.S. president nor the city are likely to be subject to the content (unless they are mentioned again separately). The most semantic approach is to annotate only the complete term. The recognition and the subsequent candidate detection process follow this assumption.

8.2.2 *Detection of Entity Candidates*

To find relevant entity candidates for the natural language terms recognized in the input texts, they must be looked up in the dictionary. The challenge is to find the terms in the knowledge base's dictionary since the nouns of the term might occur in singular or plural forms and the adjectives might occur in all possible declined forms. A dictionary constructed from anchor texts per se (see Section 8.1.1, *Extraction of Anchor Texts*) might include nouns, and combined nouns in plural and singular forms as well as adjectives in all possible declined forms. On the other hand, a dictionary constructed from alternative labels or redirects and disambiguation links (see Section 8.1.1, *Disambiguation Links and Redirects*) probably only includes nouns in singular and adjectives in basic declination. Therefore, a cleaning of the labels in the dictionary and the natural language term is essential. *Stemming* transforms words originating from the same word stem to the same principal part. Thereby, the words *run*, *runs*, and *running* are all transformed to *run*. Also, nouns in plural form are transformed to their singular form. Therefore, stemming enables the detection of entity candidates in a dictionary independent from the declination form of the natural language term in the input text. In addition to stemming, all tokens should be converted to lower case. Labels of entities are sometimes spelled with an uppercase first letter and sometimes completely in lowercase (e g. *Semantic Web* versus *semantic web*). Therefore, the conversion to lowercase overcomes the problem of inconsistent notations and case-sensitivity. The important issue is to pre-process the tokens of the natural language term so that they mimic the textual representations in the dictionary of the underlying knowledge base.

After the look-up of the natural language terms in the dictionary, every term found in the dictionary is assigned to a list containing at least one entity candidate. If the dictionary does not contain the term, it is added to the context, but rejected for annotation. The absence of the term in the dictionary does not necessarily reflect the unimportance of the term for the context, but might reveal the lack of expres-

siveness of the dictionary. Therefore the term might be important for the interpretation, although no entity is found in the dictionary.

A term possessing more than one entity candidate is considered ambiguous. The subsequent disambiguation identifies the most relevant entity among the list of candidates taking into account the present context.

8.2.3 *Co-Occurrence Analysis*

To find the contextually appropriate entity for a term, the disambiguation is processed for every entity candidate assigned to the term. The Co-Occurrence Analysis (CA) identifies the co-occurrence of the entity candidate and the other natural language terms of the context. Therefore, the disambiguation data containing co-occurrence of entities and text of the underlying knowledge base is used (see Section 8.1.3). The analysis returns a score within the range [0.0...1.0] for every entity candidate. The higher the score, the more supported is the candidate as the correct interpretation for the given context. The score is calculated as follows:

$$C(t) = \{t_j\}, j = 1...k \;\; W(uri(t)_i) = \{w_r\}, r = 1...|W(uri(t)_i)| \tag{9}$$

where t is the term currently being disambiguated. $C(t)$ is the set of terms in the context in which t has to be disambiguated. $W(uri(t)_i)$ is the set of all terms of the context occurring with the entity candidate $uri(t)_i$. To calculate the CA score, the number ($counter_{cooc_i}$) representing how often all terms of the context co-occur with the entity candidate is determined as:

$$counter_{cooc_i} = \sum_{j=1}^{k} \sum_{r=1}^{|W(uri(t)_i)|} \delta(t_j, w_r) \tag{10}$$

with

$$\delta(x, y) = \begin{cases} 1\text{: } x=y \\ 0\text{: } else \end{cases} \tag{11}$$

Finally, the CA score is calculated as follows:

$$score_{CA_i} = counter_{cooc_i} \cdot \frac{|W(uri(t)_i) \cap C(t)|}{|C(t)|} \tag{12}$$

The score is normalized with respect to the scores of all entity candidates of a term. The normalization is performed by means of the highest achieved score ($score^{max}_{CA_i}$) within the set of all entity candidates:

$$score^{normalized}_{CA_i} = \frac{score_{CA_i}}{score^{max}_{CA_i}} \tag{13}$$

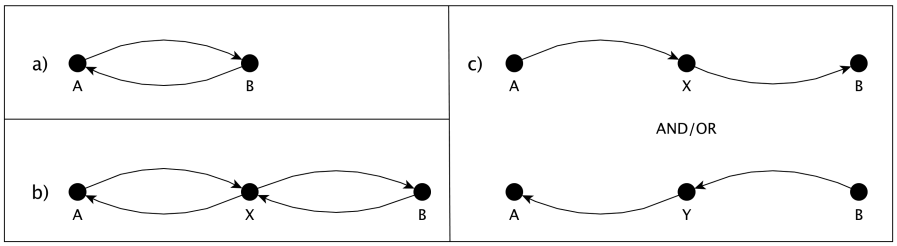


Figure 15: Three different types of links: a) direct links, b) symmetric links through same node (symlinksl2), c) unidirectional links through a node (simplelinksl2).

Thereby, a score within the range [0.0...1.0] is achieved.

8.2.4 *Link Graph Analysis*

The Link Graph Analysis (LGA) approach takes into account the relationships of entities to other entities and follows the assumption that entities that are related to each other are also linked (by means of their Wikipedia articles, see Section 8.1.3). Thus, for this analysis, the link graphs for a context's entity candidates are analyzed.

For this approach, three different link types with a maximum path length of $w = 2$ have been identified. Path length refers to the number of edges between two nodes in a graph. Therefore, a maximum path length of $w = 2$ takes into account paths where a maximum of two edges and one node is between the two nodes under consideration. The path length has been limited due to declining relation relevance for increasing path length.

The three link types represent different levels of strength with respect to their relation to the two involved entities:

- direct links
- symmetric links via a node
- unidirectional link through a node

The strongest relationship is represented by direct links where both entities point to each other. For this link type, the path length $w = 1$. In this case, the descriptive metadata mentions the respective other entity directly and a direct relationship is assumed. A less strong relationship between two entities is represented by symmetric links via a common node. This link type holds the path length $w = 2$. Both entities point to the same third entity and this entity points to both entities under consideration. Links between two entities via a node ($w = 2$) in only one direction are considered less strong than symmetric links via the same node. The link types are depicted in Figure 15 in descending order of their strength of relationship between the relevant entities.

Link types b) and c) are links with a path length of $w = 2$. That means, that these entities are linked through nodes which are also entities. For example, *Albert Einstein* and *Gottfried Leibniz* both have incoming and outgoing page links to *Berlin Academy of Sciences*, but they are not directly linked to each other within their Wikipedia articles. So, these two entities are linked via a link type b).

The LGA detects connections between the entity candidate currently under consideration and the entity candidates of the other terms in the context. A score for every link type is calculated similarly to the calculation of the score for the CA.

The connections of the entity candidate under consideration and the other entity candidates are counted. For link types b) and c) the number of different paths between two candidates are also counted. The score for direct links is calculated as follows:

$$\mathrm{counter}_{\mathrm{dlinks}_i} = \sum_{j=1}^{k} \sum_{l=1}^{m} |\mathrm{uri}(t)_i \rightarrow \mathrm{uri}(t_j)_m| \tag{14}$$

$$\mathrm{score}_{\mathrm{dlinks}_i} = \frac{|t \rightarrow t_k|}{|C(t)| \cdot \mathrm{counter}_{\mathrm{dlinks}_i}} \tag{15}$$

$\mathrm{counter}_{\mathrm{dlinks}_i}$ is the number of candidates the processed candidate ($\mathrm{uri}(t)_i$) is linked to directly. $|C(t)|$ is number of terms in the given context, and $|t \rightarrow t_k|$ is the number of terms in the given context to which the entity candidate is linked.

With this calculation, entity candidates that are linked to only one of the entity candidates of the other terms achieve high scores. Such candidates have fewer links, but these links are less ambiguous. This is because there is a link to only one of the candidates of the context term; this link is presumed to identify the relationship between the candidate under consideration and the context term uniquely. An entity candidate that is linked to more than one of the candidates of a specific term in the context is much less relevant, because these links might reveal further ambiguity.

The achieved ranking of different link options is shown in Figure 16. In the example, *uri 1* is linked to one entity candidate of every term in the context. That implies that this entity candidate is strongly related within this context. Also, relationships of this candidate to the other terms in the context are not ambiguous since the candidate is only linked to one of the candidates of each term. *uri 5* is also strongly related within this context, but the links are ambiguous since the candidate is linked to two candidates of each term. *uri 2* has the same number of links to the other entity candidates as *uri 1* in this context, but all links refer to the same term. This candidate is the least related candidate and the links are ambiguous since the candidate is linked to three different entity candidates of the same term.

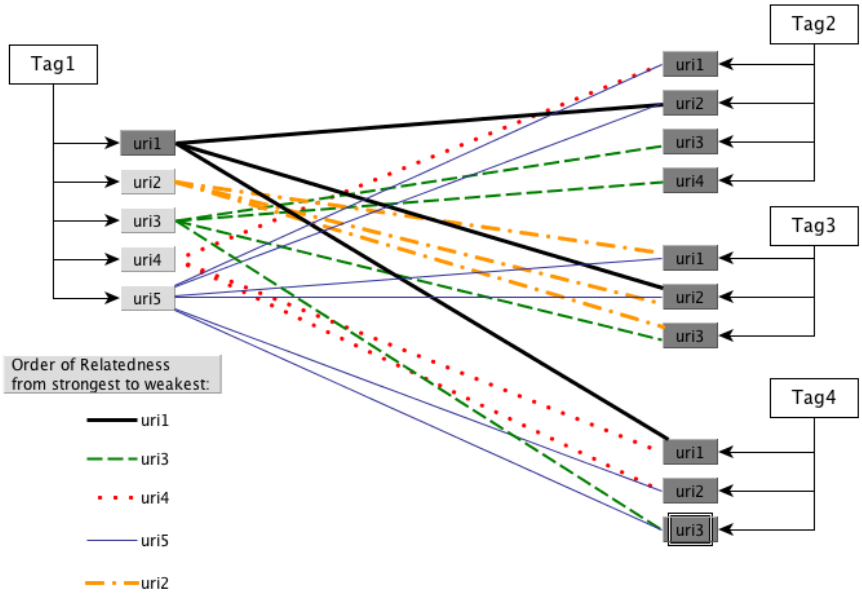


Figure 16: Relationship ranking of entity candidates via links.

The score derived by the LGA represents two characteristics of an entity candidate with respect to the given context:

- ambiguity
- integration

The two characteristics are described in detail in the following paragraphs.

AMBIGUITY The level of ambiguity is defined under the assumption that all entity candidates of a term represent different concepts. Even if the candidates do not represent completely different concepts, only one of the candidates can be the most relevant (the "best fit") for the context. Thus, the relationship of an entity candidate under consideration to more than one entity candidate of context item indicates ambiguity. The relationship of this candidate is not unique. Therefore, the score takes into account whether or not the candidate is linked uniquely to the context. Figure 16 shows the entity candidates *uri1* – *uri5* for the term *term1*. This term is under consideration and about to be interpreted. *uri1* is linked to all terms of the context (*term2*, *term3*, *term4*) and all links are unique – *uri1* is linked only to one entity candidate of each term in the context. *uri3* is also linked to all terms in the context, but the link to *term2* is ambiguous as there links to two entity candidates of *term2*. Therefore, *uri1* receives a higher score than *uri3*.

INTEGRATION The level of integration of an entity candidate under consideration is represented by the ratio of links to the terms in

the context. The score takes into account the number of terms in the context and the number of terms to which the entity candidate is linked.

The score for link type b) (symmetric links with path length $w = 2$) is calculated as follows:

$$score_{sw2_i} = \frac{|t \rightarrow t_k| \cdot p}{|C(t)| \cdot counter_{sw2}} \tag{16}$$

The score for link type c) (unidirectional link path length $w = 2$) is calculated as follows:

$$score_{lw2_i} = \frac{|t \rightarrow t_k| \cdot p}{|C(t)| \cdot counter_{lw2}} \tag{17}$$

$counter_{sw2}$ (as part of score $score_{sw2_i}$) and $counter_{lw2}$ (as part of score $score_{lw2_i}$) are identical to the calculation for direct links. They count the number of entities of the context linked to the candidate under consideration. p is the number of different paths between two linked entities. The formula above the fraction line multiplies the number of the context terms to which the candidate is linked with the number of paths. The formula under the fraction line multiplies the number of all context terms in the context with the number of context entities the candidate is linked to.

Similar to the score for the CA, all scores of the LGA are normalized with respect to the other entity candidates of a term:

$$score^{normalized}_{LGA_i} = \frac{score_{LGA_i}}{score^{max}_{LGA_i}} \tag{18}$$

Thereby, the LGA returns three different scores with a range [0...1] for every entity candidate of an ambiguous term.

8.2.5 *Coreference Analysis*

In NLP, coreference analysis refers to the multiple detection of the same named entity in a context. The challenge is to discover the referenced named entity when only mentioned by a personal pronoun. For example, in the paragraph *Mick Jagger visited Berlin. He is the front singer of the Rolling Stones.* the entity `dbp:Mick_Jagger` is mentioned twice: by his full name *Mick Jagger* and by the personal pronoun *He* at the beginning of the second sentence. For this disambiguation approach, a simplified version of coreference analysis is applied. If the same entity is a candidate for multiple different terms within the same context, it could be an indication that this might be the correct entity. Let us consider the following example paragraph: *Mick Jagger visited Berlin. Mick is the front singer of the Rolling Stones.* In this case, *Mick Jagger* is probably assigned to only one entity candi-

date. The term *Mick* might be assigned to several entity candidates, such as all persons with the first or last name Mick. The coreference analysis discover that *Mick Jagger* and *Mick* feature the same entity candidate within their candidate lists. The score of entity candidate i with respect to the entities $entities_{CI}$ contained in context CI for the coreference analysis calculates as follows:

$$score_i = \begin{cases} 1.0 : \ i \in entities_{CI} \\ 0.0 : \ else \end{cases} \tag{19}$$

8.2.6 *Additional Analysis Methods*

The previously described analysis methods are context-dependent methods. They take into account several pieces of context information to detect the most appropriate entity candidate for an ambiguous term. In some cases these context-dependent analyses are not able to make a decision among the entity candidates, because of insufficient context or missing disambiguation data. Context-independent methods focus on the common popularity of the entity candidates. Four different popularity measures have been identified:

- main label matching
- label ranking
- incoming links
- anchor-link ranking

The first two are complete entity-based popularity measures. The latter two additionally consider the term to which the entity candidates are assigned. The four different popularity measures are described in the following paragraphs.

MAIN LABEL MATCHING The main label of an entity is considered to be the most relevant and most representative label of an entity. Additionally, in community-driven datasets, such as Wikipedia, articles of popular entities have been created first and in that case the main label of the articles exactly matches the supposed main label of the entity. If an article for an entity with the same name is created later, the article's name receives a main label and an add on. The title of an article with the same label as an already existing article is extended by a disambiguating term in brackets. For example, there are several entries for *Michael Jackson* in Wikipedia. The article title of the most popular of them, referring to the late singer, is *Michael Jackson*. Another Michael Jackson who works as a writer has the article title *Michael Jackson (writer)*. Under this assumption, the term currently under consideration and the main labels of its entity candidates are analyzed. If

the main label of a candidate matches the term exactly, the candidate receives a score of $s = 1.0$. This score reflects a context-independent popularity score for the most popular entity among the candidates. Label ranking, which is discussed below, applies for terms that are only similar to the main label of the candidate.

LABEL RANKING The label ranking for alternative labels described in Section 8.1.2 can be utilized for the disambiguation process in two ways:

- Only take into account entity candidates whose label score exceeds a specific threshold.
- Use the label score as an additional heuristic comprised in the final score of the disambiguation process.

The first approach only disregards entity candidates whose label score is too dissimilar to the term under consideration. The second approach ranks the entity candidates according to the similarity of their label and the term.

INCOMING PAGE LINKS The number of incoming page links represents another entity-based popularity measure. The more entities point to an entity, the more popular this entity is considered to be. For example, for DBpedia 3.8.0, 4982 entities point to `dbp:Michael_Jackson`, but `dbp:Michael_Jackson_(writer)` features only 56 incoming links.

ANCHOR-LINK RANKING The preparation of Anchor-Link Ranking has been described in Section 8.1.2. This ranking can be used to find the most popular entity candidate for the respective term. For example, `dbp:Michael_Jackson` receives a score of $s = 0.93$ for the term *Michael Jackson* as anchor text, while `dbp:Michael_Jackson_(writer)` only receives a score of $s = 0.0093$.

8.2.7 *Final Score and Weights for Analysis Methods*

The total disambiguation score for every entity candidate is calculated from the scores the candidates achieve for the separate analysis methods. The simplest way to achieve a disambiguation score with a range of [0...1] is to calculate an averaged score from all single scores. But, as already mentioned, the analysis methods might be of different relevance for the total disambiguation score with respect to the indication for the correct entity for the term within the present context. Therefore, the assignment of a weight for the single scores might accent the more informative analysis methods. The identification of proper weights for the single scores can be performed in different ways. One

possibility is the application of a machine learning algorithm. A classifier is trained according to which score weighting achieves the best recall and/or precision on a provided benchmark dataset (see Section 10.3 for a description of entity disambiguation benchmarks). Another possibility is an empirical study. Different score weights are applied and the weight assignment achieving the best recall and/or precision is chosen. In this case, however, the score weights might be different for different benchmarks. A dataset with texts containing the most popular entities probably needs higher weights for the popularity-based analysis methods. Section 10.7.4 describes an empirical study on the determination of the weights for the scores derived from the presented analysis methods.

9

SEMANTIC ANALYSIS USING THE CONTEXT MODEL

This chapter links the previously defined *heterogenous context* and the novel *context model* for the purpose of semantic analysis on video metadata. The previously introduced *context model* (see Chapter 7) enables a ranking of context items. Thereby, a more confident disambiguation process within heterogenous contexts is pursued. The semantic analysis process under consideration of the *context model* is described in Section 9.1. The ranking of context items allows for a sequential disambiguation of items from the most confident to the least confident context item, returning evaluated entity candidates which are more and less relevant for the present context. Less relevant entities are added to a negative context. The dynamic creation of a negative context and subsequent application of the negative context in the disambiguation process is introduced in Section 9.2. Section 9.3 presents the complete semantic analysis algorithm in an overview. The chapter concludes with a discussion about complexity evaluations of the individual analysis steps in Section 9.4.

9.1 SEMANTIC ANALYSIS OF RANKED CONTEXT ITEMS

The *context model* introduced in Section 7 is used to conduct semantic analysis of video metadata for the annotation of textual information with semantic entities. In general, the semantic analysis of video metadata consists of three main steps:

- derivation of context items from video metadata and definition of contextual descriptions
- calculation of the confidence values for context items and sorting the list of context items according to their confidence values
- disambiguation all context item using dynamically created context in order of decreasing confidence values

Context items are created for all natural language terms found in the textual information of the video's metadata (see Section 7.1 for different text types and Section 8.2.1 for recognition of entities). The determination of the contextual description and the final calculation of the confidence value of a context item are described in Section 7.1. All context items are then ordered with respect to their confidence values. The disambiguation of the context items is processed in sequence from the highest to the lowest confidence value, because this

allows for a more accurate semantic analysis of context items (see Hypothesis 9.1.1). For the disambiguation process, a context is required. Taking into account the contextual description and the context items' confidence values the context for each disambiguation is created dynamically. The method is described in detail in Section 9.1.2. First, some examples of context items and their characteristics are described in Section 9.1.1.

9.1.1 *Context Item Characteristics*

Before the disambiguation process can begin, the collected context items must possess the following characteristics and features:

- a natural language term extracted from the input text under consideration for annotation
- a list of entity candidates assigned to the term according to the underlying knowledge base
- a contextual description (see Section 7.1)
- a confidence value calculated from the contextual description
- a reference to the structural segment or a timestamp for the input text within the video

To demonstrate the context item characteristics, the example from Section 7.2, depicted in Figure 11 is resumed. The characteristics of the context items are listed in Table 10. The field *Contextual Description* lists the components of an item's contextual description in the following order: text type, source, list of sources for source diversity, assigned ontology class, number of tokens. The field *Reference* refers to the video segment the context item is assigned to and corresponds to the relevance view on a context item presented in Section 7.3.2. A value of zero in this field indicates that the context item belongs to the authoritative metadata and is assigned as contextual information to the entire video.

After a context item has been disambiguated using the approach presented in Section 8.2 and the dynamically created context, the list of entity candidates is replaced by the entity determined to be the most fitting entity for the given context. The resulting entity features a disambiguation score[1]. This disambiguation score has a range of [0.0 ... 1.0] and represents a reliability value of the disambiguation process (see Section 8.2.7). The higher this score, the higher the reliability for a correct disambiguation. If an previously disambiguated context item is added to influence the disambiguation of another item, the assigned entity is used as a reference point for the context creation.

1 The score the entity candidate achieved during the disambiguation process (see Section 8.2.7)

Table 10: Characteristics of sample context items.

ID	Term	Number Of Entity Candi-dates	Contextual Description	Confidence	Reference
1	Apple	15	Keyword Authoritative (auth.) Organization one token	0.652	0
2	Operating systems	3	Text Authoritative (auth.,OCR) / two tokens	0.558	0
3	operating systems	3	Text OCR (auth.,OCR) / two tokens	0.448	655739
4	Lion	110	Text Authoritative (auth.) / one token	0.392	0
5	Jaguar	40	Text Authoritative (auth.) / one token	0.392	0
6	computer	148	Text OCR (OCR) / one token	0.282	655739

Otherwise, the entire list of all entity candidates of the context item is used for the disambiguation process. Non-ambiguous context items initially contain only one entity candidate featuring a disambiguation score of 1.0. Subject to these conditions, the following hypothesis is advanced:

Hypothesis 9.1.1. The disambiguation results of context items are improved if context items with higher confidence values are disambiguated first.

The dynamic creation of the context for the disambiguation of a context item is described in the next section.

9.1.2 *Dynamic Context Creation*

The context of a context item determines the appropriate meaning of the ambiguous textual information of the item, resulting in a single entity. The more specific the context, the higher the probability of an accurate interpretation. Typically, documents of any type are structured according to content-related segments. The more segments are contextually aggregated, the more general the contextual information is considered to be. Ambiguous textual information is difficult to disambiguate using a general context, because a general context probably contains more heterogenous information. Thus, the document should be split up into fragments of coherent content to enable the creation of more accurate contexts. Considering this presumption, the following hypothesis is put forward:

Hypothesis 9.1.2. The context of context items within a document should be restricted to segments of coherent content.

Hypotheses 9.1.1 and 9.1.2 have been confirmed by several evaluations. An annotated video metadata benchmark has been applied for the presented semantic analysis process. The results and discussion of the results are presented in Section 10.7.5.

Following Hypotheses 9.1.1 and 9.1.2, the context for the disambiguation of each context item is created dynamically. Thus, the list of context items pertaining to a context and applied for disambiguation is assembled for every context item separately. If a context item is added to a context, three criteria are applied:

- reference to the video segment,
- confidence value, and
- disambiguation score.

Obviously, the disambiguation score is only available for context items that have been previously disambiguated.

Only context items of the same segment and with a defined minimum confidence value are added to the context and thereby influence the disambiguation process. Authoritative metadata, such as titles or descriptions, are considered to be descriptive of the content of the entire document. The information given as authoritative metadata supports the interpretation of all time-referenced metadata. Therefore, the context items which originate from authoritative metadata are added as context for the disambiguation of each time-related context item. Accordingly, the exemplary context items listed in Table 10 build the following two contexts:

- Context items with ID 1,2,4,5 – the authoritative context items – are disambiguated only using authoritative context items with ID 1,2,4,5 as context (excluding themselves for their own disambiguation).
- Context items with ID 3 and 6 – originating from OCR analysis – are disambiguated using the context items from the same video segment and the authoritative context items as context. Thus, context item with ID 3 is disambiguated using context items 1,2,4,5,6 as context and the context item with ID 6 is disambiguated using context items 1,2,3,4,5 as context.

The procedure is aligned to the relevance view of a context item (see Section 7.3.2). However, a context item is only added to a context if its confidence value exceeds a certain threshold. This threshold can be set dynamically for each context item type. The threshold defines the specificity and size of the context. The lower the threshold, the more low-confident context items are added to the context. The higher the threshold, the fewer context items are added to the context. For the determination of the threshold, an exhaustive set of test runs have been performed. Section 10.7.5 presents and discusses the results of the test runs.

The same procedure applies for the disambiguation score of a previously disambiguated context item. For the dynamic context creation, the score must exceed a defined threshold. This threshold is also discussed in Section 10.7.5. By using thresholds for confidence values and disambiguation scores, the disambiguation process achieves high precision without simultaneously decreasing the recall. Using the dynamically created context, each context item is disambiguated. As the result, the context item's list of entity candidates becomes an ordered list according to the achieved disambiguation score of each individual candidate. Finally, the disambiguation process concludes with the decision for an interpretation, and thus the decision for an entity candidate. This decision can be performed in two different ways:

- Determination of all entity candidates whose disambiguation scores exceed a specific threshold, or

- Determination of only the entity candidate with the highest achieved disambiguation score.

Typically, the latter case is applied, because a unique annotation is thereby achieved. The application of a threshold usually increases the recall, but the precision decreases. If more than one entity candidate is used for the annotation, the correct entity might be included, but incorrect entities may also be chosen. The application of a specific threshold is briefly discussed in Section 10.7.7.

9.2 NEGATIVE CONTEXT CREATION

The disambiguation process introduced so far has considered only positive contexts. Thus, the scoring of the entity candidates is an additive process. Positive hints entail an increase in the disambiguation score. As discussed in Section 6.4.2, a negative context can be applied for the disambiguation process, entailing a reduction of the disambiguation score for entity candidates, if they are linked to the negative context. The negative context can be achieved by the application of either manually created black lists or dynamically extended black lists. The manual creation of black lists is briefly discussed in Section 9.2.1. The dynamic creation of negative context is enabled by the application of the context model previously introduced. This approach is presented in Section 9.2.2.

9.2.1 *Context Refinement Through Blacklists*

In the most simple way, negative context facts can be derived from black lists. These black lists can be composed of instances of ontology classes, categories or topics that should not be taken into account for entity mapping. Let us assume for example that entities in a text dealing with *Space Shuttles* are to be disambiguated. In such a case, entities which belong to the class *Punk Rock Bands* can be excluded from the disambiguation process. Therefore, all entities for *Punk Rock Bands* or the entire class can be put on a black list. Such black lists are created manually and can only be applied if the potential content of the documents to be disambiguated (or at least their general topic) is known beforehand. Unfortunately, this is a requirement which usually cannot be fulfilled when processing random web documents. In a scenario such as this, the entire knowledge base must be taken into account to map entities to semantic entities. Therefore, an approach which constructs a negative context successively during disambiguation, based on previously successfully disambiguated entities, has been developed. This approach is described in detail in the next section.

9.2.2 *Dynamic Creation of Negative Context*

Dynamic creation of a negative context is based on the assumption that a negative context is built up gradually, using excluded and eliminated entity candidates from the current disambiguation process [108]. Entities that have been an entity candidate for a natural language term in the context without achieving a sufficient score in the disambiguation process can be considered irrelevant within the current context. These rejected entities are successively added to the current negative context and serve as a basis to generate *negative topics*, i.e. topics that can be disregarded for the current context. Categories of hierarchical taxonomy systems such as Wikipedia categories[2], YAGO [110] or Umbel [9] aggregate entities that belong to similar topics. Under this assumption, previously rejected entity candidates are utilized to successively build up the current negative context.

However, this approach requires a high confidence regarding the quality of the disambiguation process. If a term is disambiguated incorrectly, potentially "wrong" entities might also be added to the negative context. This will bias both the negative and the positive contexts. Therefore, all terms within a context, i.e. context items, must be ordered according to the probability that they can be disambiguated correctly. Only then, disambiguated entities with high confidence values can be considered for context creation. The introduced context model enables the ranking of context items within a context. This in turn enables the disambiguation process to operate in sequence of the most confident to the least confident context item.

This approach is based on the presumption that entity candidates which achieve a disambiguation score of $s_{total} = 0.0$, i.e. no relationship within the context could be detected, are considered to be irrelevant within the current context. Going forward, these discarded entity candidates will be considered as part of the negative context. These entity candidates are added to the negative context[3] successively and applied for further disambiguation processes. The negative context can be considered a set of topics that are (most likely) not relevant for the present context. Therefore, the negative context can comprise not only individuals, but also more abstract or generic categories. These categories can be simply derived from the `rdf:type` respectively `dc:subject` information assigned to the negative (individual) entities. For every category assigned to a negative entity, it must be evaluated whether it is also assigned to a positive entity, i.e. an entity previously confirmed for the current context. If the category is not related to the positive context, it is added as a negative category to the negative context.

2 `http://en.wikipedia.org/wiki/Wikipedia:FAQ/Categorization`

3 Subsequently, these entities are referred to as *negative entities*, while semantic entities that are part of the positive context will be referred to as *positive entities*.

An overview of the creation of the negative context is depicted in Figure 17. In the figure, the terms *term1*, *term2*, *term3* and *term4* belong to the same context and are ordered according to the derived confidence value. The disambiguation starts with *term1* which features three entity candidates. The algorithm assigns the highest score to *uri1_1*; *uri1_3* receives a $score = 0.0$. Therefore, *uri1_3* is added to the negative context. *uri1_3* is assigned to two categories: *cat2* and *cat3*. *cat2* is also assigned to *uri1_2* which achieved a score higher than 0.0 and is therefore considered to be relevant within the context. *cat2* is therefore also considered to be relevant for the context and only *cat3* is added to the negative context. The same procedure is applied for the disambiguation of *term2*. The positive and negative context after the disambiguation of *term1* and *term2* is depicted at the bottom of Figure 17.

For this approach, DBpedia entities are used as semantic entities. Several classification hierarchies that are assigned to DBpedia entities are available, such as Wikipedia categories, YAGO or Umbel. Due to its good maintenance and cycle-free hierarchy, the YAGO categories are applied for the proposed approach.

Let us consider a disambiguation process on the authoritative metadata (title and publisher information) of the example introduced in Section 7.2. As presented in Table 6, *Apple* obtains the highest confidence value and will be disambiguated first, followed by *Operating systems*, *Lion*, and *Jaguar*. In the subsequent disambiguation process, the entity *Apple Inc.*[4] obtains the highest score compared to the other entity candidates for the term *Apple*. The categories for this entity are added to the positive categories of the current context. The entities *Apple (band)*[5] and *The Apples (Israeli)*[6] obtained a total score of $s_{total} = 0.0$ during the disambiguation. Therefore, they are added to the negative context. None of the assigned categories for these two negative entities are linked to the positive context entity *Apple Inc.* Therefore, the categories of these discarded entities are also added to the negative categories for the current context. The context following disambiguation of the term *Apple* is depicted in Table 11. With every disambiguated term, the negative context and the sets of positive and negative categories grow and can be applied for subsequent disambiguation processes. The influence of the negative context and negative categories on the disambiguation process is described in the next section.

In addition to the heuristics introduced in Section 8.2, an analysis method has been developed that makes use of the negative context information. The integration of negative context information in the disambiguation process is presented in the next section.

4 http://dbpedia.org/resource/Apple_Inc.
5 http://dbpedia.org/resource/Apple_(band)
6 http://dbpedia.org/resource/The_Apples_(Israeli)

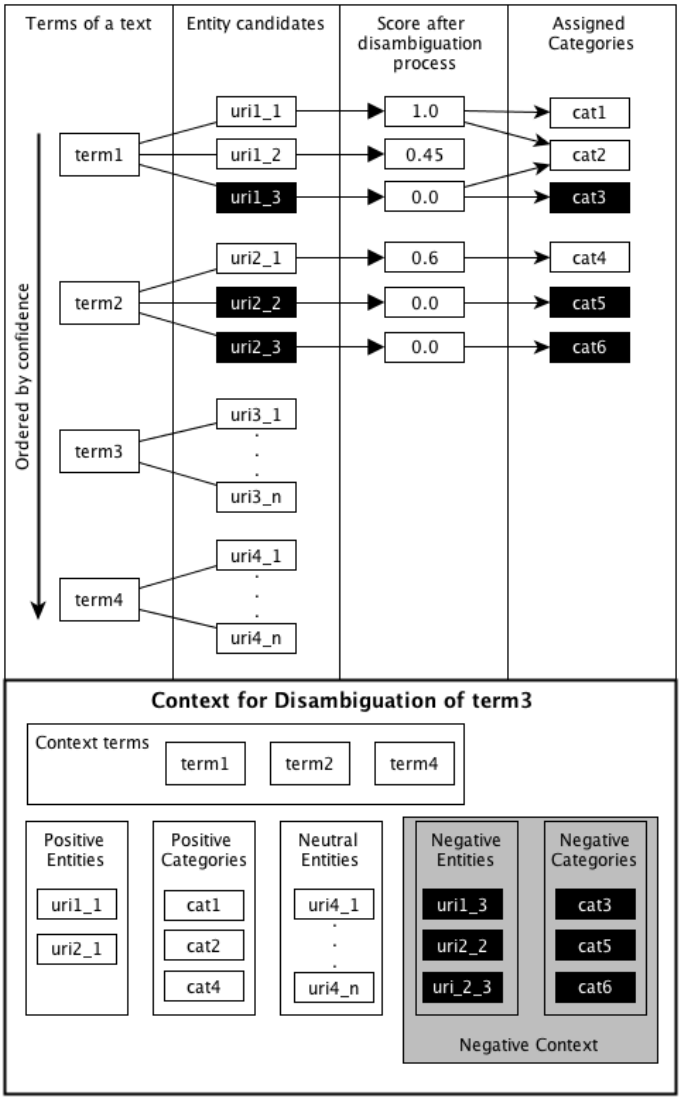


Figure 17: Creation of a negative context in the disambiguation process.

Table 11: Context for the example sentence after disambiguation of the term "Apple".

Positive Entities	*Apple Inc.* for term *Apple*
Positive Terms	*Operating systems* (1 candidate) *Lion* (88 candidates) *Jaguar* (58 candidates)
Positive Categories	yago:Company yago:CompaniesEstablishedIn1976 yago:SteveJobs yago:NetworkingHardwareCompanies
Negative Categories	yago:EnglishRockMusicGroups yago:MusicalGroupsEstablishedIn1968 yago:1960sMusicGroups

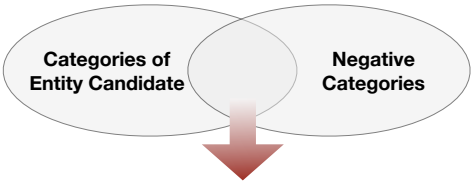
9.2.3 *Calculation of the Negative Score*

Entity candidates are first checked for whether the current negative context already contains the entity. If so, the entity candidate might not be considered for further disambiguation of the context item under consideration.

If the candidate is not part of the current negative context, the categories assigned to the entity are retrieved. The resulting set of categories and the set of negative categories in the negative context are examined for intersection. The negative score assigned to an entity depends on the size of this intersection and on how specific or general the categories of the intersection are. The specificity of a category can be derived from the category's tree depth within the classification hierarchy. The higher the tree depth, the more specific the category [52]. More general categories usually contain a higher number of entities and the relevance of this category for the entities to be disambiguated is lower than for more specific categories containing only a few entities. Thereby, a category can be weighted taking into account its significance for the entity candidate.

The weight is calculated from the logarithm of the tree depth, proportional to the logarithm of the maximum tree depth within the considered classification system[7]. The logarithm is applied because it is assumed that the degree of abstraction is not distributed linearly along the path from the root node to a leaf node. The degree of abstraction decreases very close to the root node. The application of the logarithm for the calculation of the weight of a category reflects this assumption. Thereby, a weight for the category regarding the entity candidate is achieved within the range [0.0...1.0]. For the total neg-

7 For the YAGO classification system, a maximum tree depth of 18 has been calculated.



Category	Tree Depth	Weight
cat_1	t_1	$weight_{cat_1} = \frac{log(t_1)}{log(t_{max})}$
⋮	⋮	⋮
cat_n	t_n	$weight_{cat_n} = \frac{log(t_n)}{log(t_{max})}$

Figure 18: Intersection of negative and entity candidate categories and influence of tree depth.

ative score, the weights of all categories in the intersection are first summed and then divided by the intersection size where I denotes the intersection of the categories in the current negative context and the categories assigned to the entity candidate:

$$s_{negative} = \frac{\sum_{i=1}^{n} \frac{log(t_i)}{log(t_{max})}}{|I|} \tag{20}$$

Here, I contains categories c_i, $c_i \in I$, for $1 \leqslant i \leqslant n$. t_i is the tree depth assigned to category c_i, and t_{max} denotes the maximum tree depth of the applied category hierarchy.

A sketch of the approach is depicted in Figure 18.

TOTAL SCORE The total score for an entity candidate, taking into account the negative context, can be calculated in two different ways. Either the *total positive score* previously achieved by an entity candidate is set to zero as soon as the *negative score* is unequal zero, or the *total positive score* is reduced by the *total negative score* resulting in a final score within the interval [-1.0...1.0]. For the latter case, the negative score can also be weighted to decrease or increase the influence of the negative context on the total score of an entity candidate. Both calculation approaches have been evaluated. The achieved results and the discussion of the results are explained in Section 10.7.7.

FINAL DECISION After the scoring process, the entity candidate that has achieved the highest total score is typically chosen as the prospectively correct disambiguation for the term of the context item under consideration (see Section 9.1.2). Sometimes, none of the entity candidates has achieved a positive score, i.e. all entity candidates hold

a total score $s_{total} = 0.0$ or multiple candidates have achieved the same disambiguation score. There may be several reasons for this:

- Multiple candidates that hold the same disambiguation score ($\leqslant 0.0$) are relevant within the given context.
- All candidates hold the disambiguation score ($score = 0.0$) and are irrelevant within the given context.
- The context or the entity candidates provide too little information, so that the context-based analysis cannot decide appropriately.

In these cases, an additional context-independent heuristic might be applied. Entity-based popularity measures such as incoming links, or anchor-link ranking can identify the most popular entity of all candidates. In this way, the recall is often boosted. However, it is important to choose the most popular entity only from the set of entity candidates that have *not* achieved a negative score with respect to the negative context.

9.3 ALGORITHM OVERVIEW

The proposed method of semantic analysis of video metadata is assembled by a number of single processes. The complete workflow is depicted in Figure 19. The main processes of the entire workflow are marked in the figure:

1. All metadata for a video are collected.
2. For all metadata items, named entities and key terms are identified.
3. For all identified entities within the metadata item: entity candidates are looked up in the knowledge base, confidence values are derived for the determined contextual descriptions, and a context item is created that contains all the previously gathered information.
4. All context items are sorted according to their confidence values. The subsequent analysis of the context items is performed in descendent order of the assigned confidence values.
5. The context for the disambiguation process is created dynamically for each context item separately.
6. Disambiguation analyses are performed and the disambiguation scores are identified for all entity candidates of a context item.

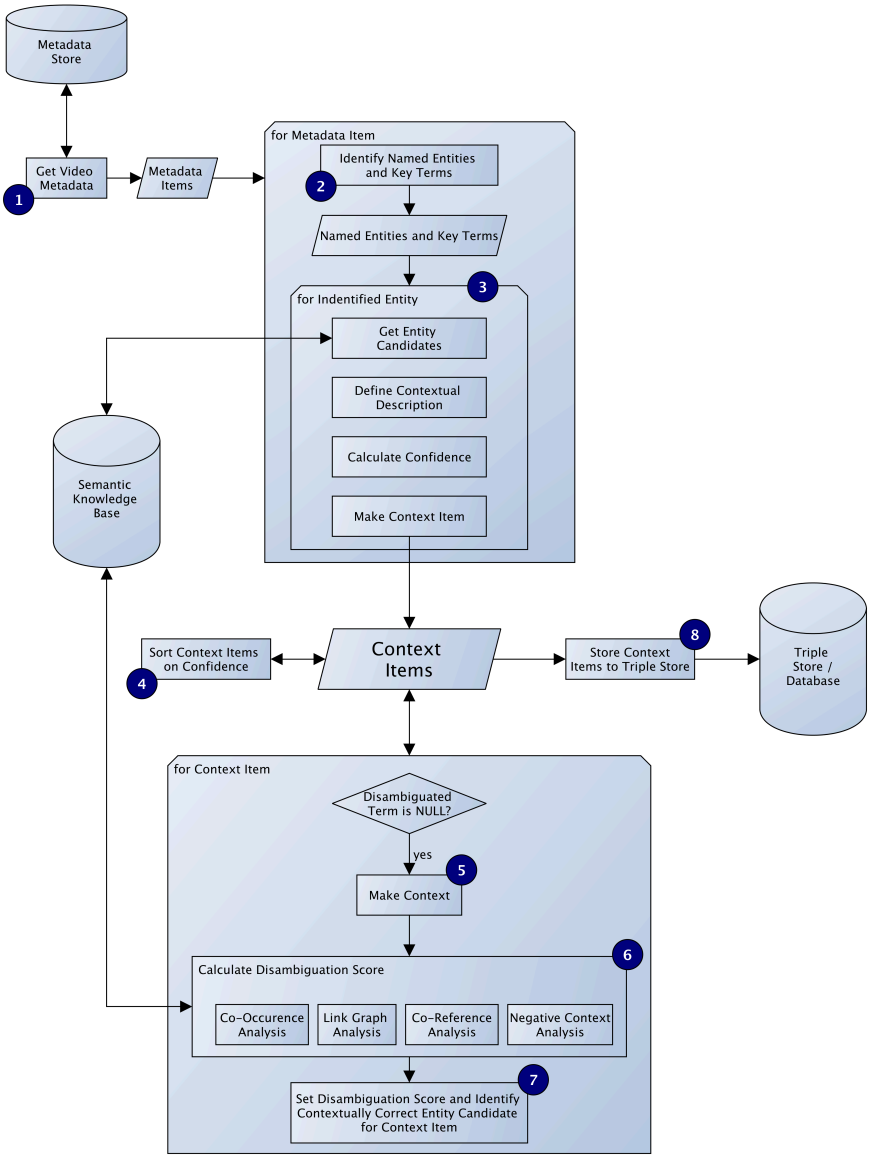


Figure 19: Complete workflow of the proposed method of semantic analysis of video metadata.

7. The contextually most relevant entity is identified by means of the highest disambiguation score. This entity candidate is assigned to the context item as an annotated entity for the context item's textual term.

8. All context items and their annotated entities are stored in a triple store or database.

The following section gives a brief overview of the computational analysis of the complete algorithm.

9.4 COMPLEXITY EXAMINATION

The complexity of the proposed semantic analysis algorithm is briefly discussed in this section. Complexity estimations for each method's individual steps are hereby examined. In general, the complexity of time and the complexity of space must be considered. The main factor for the complexity estimation is the size of the underlying knowledge base $|K|$. Its size can be estimated by the number of contained entities n including their incoming and outgoing links which results to $n^2 \hat{=} |K|$.

9.4.1 *Complexity of Space*

Generally speaking, the analysis process holds all information of a context in a cache to process the metadata. At most, the context might contain all entities contained in the underlying knowledge base. The context information is then used throughout the entire process. The Link Graph Analysis requires information about incoming and outgoing links of all entities in the knowledge base. The knowledge base contains a maximum of $n \cdot n$ links. The complexity of space of the presented algorithm is therefore determined as $O(n^2)$. The implemented sorting algorithm (Step 4 in the algorithm overview in the previous section) uses *merge sort* and needs additional space to sort the context items which pertain to a context. In the worst case the context contains n context items. The complexity of space of the sorting algorithm is therefore determined as $O(n \cdot \log(n))$ in the worst case and as $O(n)$ in the best case.

9.4.2 *Complexity of Time*

The complexity of the algorithm with respect to time is estimated for all individual steps depicted in Figure 19. The assumption for all separate steps is that a context consists of a context item for each entity contained in the knowledge base. Therefore, the worst case size of a context is determined to n.

STEP 1: COLLECTION OF METADATA At maximum, metadata for each entity contained in the knowledge base are collected which results in n metadata items. Therefore, this step takes $O(n)$ time.

STEP 2: IDENTIFICATION OF ENTITIES Similar to Step 1, only n different entities can be identified in the metadata, resulting in n context items. The identification of entities takes $O(n)$ time.

STEP 3: CREATION OF CONTEXT ITEMS For each context item identified in the previous step, entity candidates must be looked up in the dictionary. The dictionary contains at most n entities and k labels for each entity. The labels of the dictionary are stored as b-tree index structure. The look-up in such a structure for $k \cdot n$, with $k \in \mathbb{N}$, entries costs $O(\log(n))$ time for one context item. Therefore, the candidate look-up for n context items takes $O(n \cdot \log(n))$ time. For the calculation of the confidence values, all context items are processed once which takes $O(n)$ time.

STEP 4: SORTING OF THE CONTEXT ITEMS Sorting n context items using a merge sort algorithm costs $O(n \cdot \log(n))$.

STEP 5: CREATION OF CONTEXT For each context item, the context is created separately. All context items are checked for various characteristics before being added or rejected for a context for the disambiguation of the target context item. Therefore, the creation of the contexts for n context items costs $O(n^2)$ time.

STEP 6: APPLICATION OF THE DISAMBIGUATION Each of the n context items can feature at most n entity candidates. All candidates must be checked in the following analyses:

- For the co-occurrence analysis, comparisons with at most $n-1$ terms of the context items in the context are required, because the context consists of at most $n-1$ items. The co-occurrence analysis costs $O(n^2(n-1)) = O(n^3 - n)$ time.
- For the link graph analysis for direct links, all n entity candidates of all n context items must be compared with n entity candidates of $n-1$ context items. The link graph analysis for direct links takes $O(n^2(n^2-n)) = O(n^4 - n^3)$ time.
- For the link graph analysis for links via a node (path length $w = 2$), all n entity candidates of all n context items must be compared with n entity candidates of $n-1$ context items and the n entity candidates can be linked with at most n other entities. The complexity of time is therefore calculated as $O(n \cdot n \cdot (n-1) \cdot n \cdot n) = O(n^5 - n^4)$.

- For the coreference analysis, all n entity candidates of all n context items must be compared with n entity candidates of $n-1$ context items. This step takes $O(n^4 - n^3)$ time.

STEP 7: IDENTIFICATION OF CONTEXTUALLY MOST RELEVANT ENTITY CANDIDATE All n entity candidates of n context items must be checked for the achieved score. The complexity of time is calculated as $O(n^2)$.

STEP 8: STORAGE OF CONTEXT ITEMS Finally, the context items are stored and annotated with the contextually most relevant entity candidate which takes $O(n)$ time.

The presented approach can be considered a "greedy" analysis process. The complexity mainly depends on the size of the underlying knowledge base. The more entities under consideration in the knowledge base, the more complex the algorithm with respect to time and space.

Part III

EVALUATION AND APPLICATIONS

10 EVALUATION

This chapter outlines extensive evaluations which were performed to survey the quality of the annotations provided by the presented semantic analysis method. Evaluation objectives and requirements are introduced in Section 10.1. The applied evaluation measures are described in Section 10.2. For the actual evaluation of the presented algorithms and context model, several benchmarks and entity dictionaries have been surveyed. Results are presented in Sections 10.3 through 10.6. The chapter first introduces five different benchmark datasets for the purpose of the evaluation of semantic analysis engines as well as four dictionaries for the entity look-up (see Section 8.1.1 for descriptions of different approaches). The benchmarks are described in Section 10.3 and furthermore analyzed for structure and type information of the annotated entities (see Section 10.4). Section 10.5 introduces four different dictionaries. Three of them are freely available, while the fourth has been created according to the approach described in Section 8.1.1's *Redirects and Disambiguation Links*. Extensive statistics have been calculated using the four dictionaries on the five benchmarks. The statistics and the discussion of the results are presented in Section 10.6.

For simplification, the proposed disambiguation process on video metadata applying the context model is subsequently referred to as *conTagger*. The evaluation of the context model and the developed analyses for disambiguation of textual information are presented in Section 10.7. Section 10.8 briefly summarizes all evaluation results presented in this chapter. The discussion in Section 10.9 concludes the chapter.

10.1 EVALUATION OBJECTIVES AND REQUIREMENTS

Evaluation of algorithms or models can focus on either the performance with respect to the quality of the achieved results, or the consumption of resources and runtime. The evaluations described in the following sections solely concentrate on the quality of the results. For the comparison of several systems of the same task, two different approaches might be considered. On the one hand, results of automatic systems can be compared to results provided by human annotators. For this approach, a gold standard is required, but the creation of an independent and objectively annotated benchmark is time-consuming. On the other hand, the results of different approaches can be compared on the basis of the maximum agreement of all ap-

proaches on the same input data. This approach does not require a benchmark, but the quality of the results can only be compared according to a common baseline of the applied extraction methods. Gangemi utilized the latter approach to compare several knowledge extraction tools [36]. For the evaluation of the proposed algorithms the first approach is applied. Therefore, the quality of the results of semantic annotation services depends on several elements:

- the structure and type of the applied benchmark
- the dictionary for look-up of entity candidates for textual terms
- the algorithm for detection of named entities in continuous text
- the disambiguation algorithm

The following sections present evaluations according to these four elements.

EVALUATION MEASURES The evaluation results are presented using the common evaluation measures of recall, precision, F_1-measure and accuracy. The measures are briefly described in Section 10.2. Additionally, the measure *margin* is introduced. This measure has been applied to evaluate the quality of the results utilizing the negative context.

BENCHMARKS AND DICTIONARIES Several benchmarks and dictionaries have been evaluated. Evaluations on the benchmarks and dictionaries are described in Sections 10.3 and 10.5. The actual evaluations of the algorithms and the context model have been applied to the benchmarks most relevant for the respective purpose and utilizing the best fitting dictionary.

EVALUATION OF ALGORITHMS The tag processing method presented in Section 8.2.1 has been evaluated using a specific benchmark consisting of one hundred contexts containing user-generated tags and their corresponding semantic annotations. The benchmark is described in Section 10.3.2 and the evaluation results are presented in Section 10.7.1.

The algorithm for the detection of named entities in continuous text (see Section 8.2.1) has been evaluated using two of the five presented benchmarks. Results are described and discussed in Section 10.7.2.

The disambiguation process presented in Section 8.2 has been evaluated against DBpedia Spotlight using the benchmark the developers of Spotlight introduced in [75] to evaluate their approach. The benchmark is described in Section 10.3.3 and the evaluation results are presented in Section 10.7.3.

The semantic analysis process introduced in Sections 8.2 and 9.1 using the context description model (see Section 7) has been evaluated using two of the presented benchmarks and against five other approaches. The evaluation results are described in Section 10.7.5.

The results of the evaluation of the influence of a negative context on a disambiguation process are presented in Section 10.7.7. The approach has been evaluated on two different datasets.

Section 10.8 summarizes the statistics and results performed to evaluate the presented semantic analysis process.

Section 10.9 discusses the achieved evaluation results.

10.2 EVALUATION MEASURES

For the evaluation of the introduced approaches, several measures are applied. In *Information Retrieval*, recall and precision constitute established measures to present the results of an algorithm and compare them to other approaches. Both measures have been applied for the presented evaluations and are briefly described in Sections 10.2.1 and 10.2.2. The measure of accuracy is described in Section 10.2.4. Additionally, the margin of evaluation results has been investigated for the application of negative context. Margin as an evaluation measure is described in Section 10.2.5.

10.2.1 *Recall*

Recall denotes the proportional amount of relevant resources from all relevant resources. It is calculated as follows:

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

TP (true positives) denotes the number of relevant resources within the list of all retrieved resources. FN (false negatives) denotes the number of relevant resources that have not been retrieved by the algorithm.

10.2.2 *Precision*

Precision denotes the proportional amount of relevant resources from all retrieved resources. It is calculated as follows:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

Again, TP denotes the number of relevant resources within the list of all retrieved resources. FP (false positives) denotes the number of retrieved resources that are not relevant.

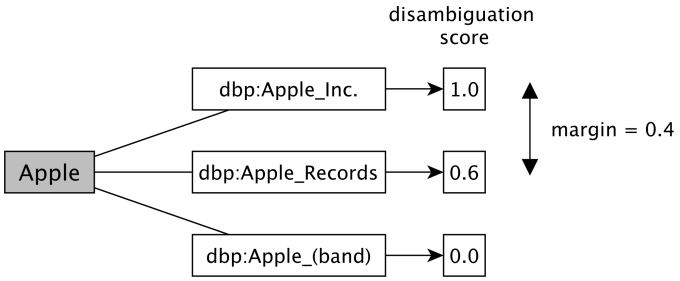


Figure 20: Margin of disambiguation scores of entity candidates after disambiguation.

10.2.3 F_1-*Measure*

The F_1-measure (also F_1-score) constitutes the harmonic mean of precision and recall. It is considered to be the evenly weighted average of precision and recall. The F_1-score is calculated in the following way:

$$F_1 = 2 \cdot \frac{recall \cdot precision}{recall + precision}$$

10.2.4 *Accuracy*

The accuracy of an algorithm takes into account the correct absence of a result – the true negatives (TN). It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

In addition to recall and precision, the accuracy speaks to the quality of the actual annotated results. For the presented evaluation, this measure is applied to evaluate the quality of the annotations with respect to the position and length of the annotated segment in the source text.

10.2.5 *Margin*

Semantic analysis algorithms usually return a scored list of entity candidates for an ambiguous term. Thus, every entity candidate possesses a score received during the disambiguation process. The distance between the highest and the second highest achieved score within the scores of all entity candidates denotes the margin (see Figure 20). Thus, the margin denotes the level of reliability of the disambiguation algorithm. If the scores are very close, the uncertainty of the disambiguation is revealed. The entity with the highest score might be the best fit within the context, but the entity candidate possessing the second highest score might also be strongly related to the context.

Therefore, the higher the margin between the scores, the higher the reliability of the disambiguation process.

10.3 BENCHMARKS

In the following section, several benchmarks for the purpose of semantic analysis and NED evaluation are presented. Due to the specific research field of heterogenous contexts and the absence of a semantically annotated dataset for this purpose, a dataset containing textual information of different characteristics and different provenance has been created. This dataset is described in Section 10.3.1. To also apply the presented approach on an independent dataset, the dataset originally published for the evaluation of *DBpedia Spotlight* has been applied. This dataset is described in Section 10.3.3. The tag processing approach described in Section 8.2.1's *Tag processing* has been evaluated using a specific dataset containing only tag groups. This dataset is described in Section 10.3.2. Additionally, two freely available benchmark datasets are introduced in Sections 10.3.4 and 10.3.5.

10.3.1 *Video Metadata Benchmark*

The evaluation of *conTagger* requires a dataset consisting of different types of video metadata including the correct entities assigned to all available textual information. As far as the author knows, no dataset of that structure and for that purpose exists. Therefore, a new dataset containing annotated video metadata has been created in order to evaluate the presented approach. The dataset has been annotated by four different domain experts.

The evaluation dataset consists of metadata from five videos. The videos are live recordings of TED[1] conference talks covering the topics of physics, biology, psychology, sociology and history science. The videos have been chosen for their diverse subject matter and because they contain a sufficient amount of displayed text for information extraction by OCR. All videos contain spoken text for information extraction by ASR.

The metadata for each video consist of authoritative metadata (including title, speaker, providing organization, subject, keywords, descriptive text and a Wikipedia text corresponding to the speaker), user-generated tags and automatically extracted text from OCR and ASR. The videos have been partitioned into content-related video segments via automatic scene cut detection (see Section 2.4.1). The time-related metadata (tags, ASR and OCR) are assigned to the respective video segments. Overall, the dataset consists of 822 metadata items,

1 http://www.ted.com

where an item can be a key term or continuous text consisting of up to almost 1000 words[2].

10.3.2 *Tag Benchmark*

The tag benchmark has been created for the evaluation of the tag processing approach (see Section 8.2.1). The data set consists of one hundred sample contexts. Fifty contexts are formed from tags that have been tagged at the same timestamp in a video. The other fifty sample contexts have been tagged in the same segment of a video. Each sample context provides a list of three to 12 single tags. Within a context, the combination of tags in every order might represent an entity, because the underlying tagging system does not allow for tags to consist of multiple words. Overall, the dataset contains 505 annotated tags or groups of tags resulting in 428 distinct entities.

10.3.3 *Spotlight Benchmark*

Mendes et al. published a dataset for the evaluation of the semantic annotation tool *DBpedia Spotlight* [75]. This test set consists of ten New York Times text paragraphs and a ground truth containing 249 extracted DBpedia entities.

10.3.4 *KORE 50 Benchmark*

KORE 50 [51] is a subset of the larger AIDA corpus [50], which is based on the dataset of the CoNLL 2003 NER task. The dataset aims to capture difficult to disambiguate mentions of entities and contains 50 sentences from different domains, such as music, celebrities and business. It is provided in a clear TSV format.

10.3.5 *Wikilinks Benchmark*

The Wikilinks Corpus [100] was introduced by Google in March 2013. The corpus collects hyperlinks to Wikipedia articles gathered from over three million web sites. It has been transformed to RDF format using the NLP Interchange Format (NIF) by Hellmann et al. [47]. The corpus is divided into 68 RDF dump files, from which the first one has been used for dictionary statistics (see Section 10.5). This benchmark cannot be considered as a gold standard. In some cases, mentions are linked to broken URLs, redirects or semantically wrong entities. This issue is discussed further in Section 10.5.

2 For downloading the dataset and the ground truth, please see the readme file at `http://tinyurl.com/cztyayu`

10.4 BENCHMARK STATISTICS

The five benchmark datasets under consideration cover different domains; though three datasets originate from authentic corpora, varying portions have been selected and different types of entities have been annotated. The statistics for the three benchmarks of Sections 10.3.3, 10.3.4 and 10.3.5 have been calculated by Steinmetz et al. (see [109]). They are shown in Table 12. Additionally, the benchmark statistics for the video dataset and the tag dataset are shown in Table 13.

The DBpedia Spotlight evaluation dataset contains 60 natural language sentences from ten different documents with 249 annotated DBpedia entities, wherein about 10% of the annotated entities are locations. About 80% of the annotated entities are not associated with type information based on the DBpedia ontology (except for `owl:Thing`).

The KORE 50 dataset contains 144 annotations which mostly refer to agents (74 times `dbo:Person` and 28 times `dbo:Organisation`). Only a relatively small amount (18.5%) of annotated entities do not provide any type information in DBpedia. The context for the annotated entities in the KORE 50 dataset is limited to (relatively) short sentences.

The Wikilinks dataset covers almost every possible domain. Its sheer size allows the extraction of sub-benchmarks for specific designated domains; for example there are about 281,000 mentions of 8,594 different diseases [109]. For each annotation of the Wikilinks dataset, the original website is named, which allows recovery of the full document contexts for the annotations, though they are not contained in the NIF resource so far.

The video metadata benchmark and the tag dataset represent benchmarks for the special purpose of semantic video analysis. The video metadata benchmark contains metadata of four different source types: authoritative, tags, OCR and ASR. As shown in Table 13, places and persons are mainly represented in OCR data and tags. However, only a small number of the annotated entities are typed differently from `owl:Thing` across the annotations of all source types (between 14% and 18%).

Overall, only the KORE 50 benchmark provides more typed than untyped annotations. For the video metadata dataset, tag dataset, Spotlight benchmark and Wikilinks benchmark, between 66% and 87% of the annotations are untyped. For the KORE 50 benchmark, the inverse ratio applies.

For the evaluation of the semantic analysis approach under consideration of the context model, the video metadata benchmark has been applied, because it is the only benchmark providing metadata which originates from multiple sources. The Spotlight benchmark has been used to show results of the presented approach on an independent benchmark and the tag benchmark has been used to evaluate the tag

Table 12: Distribution of DBpedia types in Benchmark Datasets KORE 50, Wikilinks and Spotlight (See [109] for a more detailed view).

Class	Spotlight	KORE 50	Wikilinks
Total	249	130	2,228,049
Untyped	79.9%	18.5%	66.5%
Activity	<1%	–	<1%
- Sport	<1%	–	<1%
Agent	2.4%	66.9%	18.9%
- Organisation	<1%	18.5%	5.3%
- - Company	<1%	9.2%	1.8%
- - SportsTeam	–	7.7%	<1%
- - - SoccerClub	–	7.7%	<1%
- Person	2.0%	48.5%	13.6%
- - Artist	–	17.7%	3.4%
- - - MusicalArtist	–	17.7%	1.8%
- - Athlete	–	6.9%	1.2%
- - - SoccerPlayer	–	5.4%	<1%
- - Officeholder	<1%	4.6%	1.1%
Colour	1.6%	–	<1%
Disease	1.6%	–	<1%
EthnicGroup	1.2%	–	<1%
Event	1.2%	–	1.0%
Place	10.4%	10.8%	9.6%
- ArchitecturalStructure	2.0%	3.1%	1.8%
- - Infrastructure	1.6%	<1%	<1%
- PopulatedPlace	7.2%	5.4%	5.1%
- - Country	3.6%	–	<1%
- - Region	<1%	–	<1%
- - Settlement	2.4%	3.8%	3.8%
- - - City	1.6%	2.3%	<1%
Work	<1%	6.2%	6.9%
- Film	–	–	1.9%
- MusicalWork	<1%	3.1%	1.2%
- - Album	<1%	3.1%	<1%
Year	<1%	–	<1%

processing method. The KORE 50 benchmark is a very specific benchmark, because it contains mainly first names of persons and has been created to evaluate AIDA. The Wikilinks benchmark cannot be considered as ground truth, because many of the annotations are incorrect. Therefore, these two datasets have not been used for the evaluation of the presented approaches.

Table 13: Distribution of DBpedia types in Video Metadata Benchmark

Class	Video Dataset				Tag Dataset
	Autho-ritative	Tags	ASR	OCR	
Total	221	263	784	107	428
Untyped	81.9%	76.81%	87.24%	75,7%	83.6%
Activity	–	–	<1%	–	–
Agent	6.79%	9.88%	4.47%	10.28%	8.70%
- Organisation	3.17%	2.66%	1.28%	3.74%	<1%
- - Legislature	–	–	<1%	–	–
- - MilitaryUnit	–	<1%	<1%	<1%	–
- - Company	1.81%	<1%	<1%	–	<1%
- - Band	–	<1%	<1%	–	<1%
- - EducationalInstitution	1.36%	<1%	<1%	2.80%	<1%
- Person	3.62%	7.22%	3.19%	6.54%	8.64%
- - Journalist	–	–	<1%	–	–
- - FictionalCharacter	–	<1%	<1%	–	<1%
- - Scientist	3.62%	3.04%	1.40%	4.67%	2.8%
- - OfficeHolder	–	<1%	<1%	–	<1%
- - Philosopher	–	1.14%	<1%	<1%	1.4%
- - Artist	–	1.52%	<1%	1.87%	<1%
AnatomicalStructure	<1%	1.52%	1.15%	–	<1%
Colour	–	1.52%	<1%	–	–
Currency	–	–	<1%	–	–
Disease	<1%	<1%	1.28%	<1%	<1%
Place	2.71%	3.42%	2.17%	8.41%	1.40%
- HistoricPlace	–	<1%	–	<1%	–
- NaturalPlace	<1%	<1%	<1%	–	–
- - BodyOfWater	<1%	<1%	<1%	–	–
- - - Lake	<1%	<1%	<1%	–	–
- PopulatedPlace	2.26%	2.66%	1.91%	7.48%	1.4%
- - Region	–	–	<1%	1.87%	<1%
- - - AdministrativeRegion	–	–	<1%	1.87%	<1%
- - Island	–	–	<1%	–	<1%
- - Country	1.36%	1.52%	1.02%	1.87%	<1%
- - Continent	<1%	<1%	<1%	<1%	–
- - Settlement	<1%	<1%	<1%	2.80%	–
Species	4.07%	4.18%	2.30%	3.74%	<1%
- Eukaryote	3.17%	3.80%	2.04%	2.80%	<1%
- - Plant	–	<1%	–	–	–
- - Animal	3.17%	3.42%	2.04%	2.80%	<1%
Work	2.26%	<1%	<1%	–	<1%

10.5 DICTIONARIES

Dictionaries contain associations that map strings (surface forms) to entities represented by Wikipedia articles or DBpedia concepts. Typically, dictionaries are applied by semantic analysis systems in an early step to find candidates for terms in natural language texts. In a further (disambiguation) step, the actual correct entity must be selected from all candidates.

10.5.1 *Spotlight Dictionary*

The DBpedia Lexicalizations dataset [75] has been extracted from Wikipedia interwiki links. It contains anchor texts, the so-called surface form, along with their respective destination articles. Overall, there are two million entries in the DBpedia Lexicalizations dataset. For each combination, the conditional probabilities, $P(uri|surfaceform)$[3] and $P(surfaceform|uri)$, and the pointwise mutual information value (PMI) are given. $P(uri|surfaceform)$ represents the probability that for a given `surfaceform` as anchor text, an `uri` (which represents an entity) is linked. $P(surfaceform|uri)$ represents the probability that for a given `uri` a `surfaceform` is used as anchor text. PMI represents the probability of the joint co-occurrence of `uri` and `surfaceform` under the assumption that both are independent from each other. Subsequently, this dictionary is referred to as *SPL*.

10.5.2 *Google Cross-Wiki Dictionary*

Google has released a similar but much larger dataset: Crosswiki [103]. The Crosswiki dictionary has been built at webscale and includes 378 million entries. This dictionary is subsequently referred to as *GCW*. Similar to the SPL dataset, the probability $P(uri|surfaceform)$ has been calculated and is available in the dictionary. This probability is used for the experiments described in Section 10.6.

10.5.3 *AIDA Means Dictionary*

The *AIDA Means* dictionary is an extended version of the YAGO2[4] means relation. The YAGO2 means relation is harvested from disambiguation pages, redirects and links in Wikipedia [126]. Unfortunately, there is no information given as to what the extension includes exactly. The AIDA Means dictionary contains approximately 18 million entries. Subsequently, this dictionary is referred to as *AIDA*.

3 The measure is used later for the experiments as Anchor-Link-Probability (see Section 10.6).

4 `http://www.yago-knowledge.org/`

10.5.4 *DBpedia-Based Dictionary*

In addition to the three already existing dictionaries described above, a dictionary according to the approach described in Section 8.1.1's *Redirects and Disambiguation Links* has been created. Initially, except for the elimination of bracket terms (e. g. the label *Berlin (2009 film)* is converted to *Berlin* by removing the brackets and the term within them), no further preprocessing has been performed on this dictionary. Thus, all labels are presented with original case sensitivity. Further evaluation of this issue is provided in Section 10.9. This dictionary is subsequently referred to as *RDM*.

10.6 DICTIONARY STATISTICS

The benchmarks described in Section 10.3 are constructed to evaluate semantic analysis algorithms. The evaluation results of an approach are not only dependent on the actual algorithm used to disambiguate ambiguous mentions but also on the structure of the benchmark and the underlying dictionary utilized to determine entity candidates for a mention. A *mention mapping* or *mapped mention* refers to a mention of a benchmark that is assigned to one or more entity candidates of the used dictionary.

The experiments described in the following section have been conducted to identify several characteristics of the introduced dictionaries as well as to consolidate assumptions about the structure of the benchmarks. For performance reasons only a subset of the Wikilinks benchmark has been used in the experiments. A dump file containing 494,512 annotations and 192,008 distinct mentions and assigned entities has been used for the subset.

10.6.1 *Experiments*

MAPPING COVERAGE First, the coverage of mention mappings is calculated. All annotated entity mentions from the benchmarks are looked up in the four different dictionaries. If at least one entity candidate for the mention is found in the dictionary, a counter is incremented. This measure is an indicator of the expressiveness and versatility of the dictionary.

ENTITY CANDIDATE COUNT For all mapped mentions, the number of entity candidates found in the respective dictionary is added up. The number of entity candidates corresponds to the level of ambiguity of the mention and can be considered an indicator for the level of difficulty of the subsequent disambiguation process.

MAXIMUM RECALL The list of entity candidates for all mapped mentions are looked up whether or not the annotated entity (from the benchmark) is included. Disambiguation depends upon it being contained in the list. Thus, this measure predicts the maximum possible recall using the respective dictionary on the benchmark.

RECALL AND PRECISION ACHIEVED BY POPULARITY For WSD, once entity candidates are determined for the mentions, a subsequent disambiguation process tries to detect the most relevant entity of all candidates according to the given context. For this experiment, the disambiguation process is simplified: the most popular entity among the available candidates is chosen as correct disambiguation. To determine the popularity of the entity candidates, three different measures are considered:

- Incoming Page Links of entity candidates
- Anchor-Link probability within web document corpus
- Anchor-Link probability within Wikipedia corpus

The first measure is a simple entity-based popularity measure. The popularity is defined according to the number of incoming Wikipedia page links. The more links point to an entity, the more popular the entity. Anchor-Link probability defines the probability of a linked entity for a given anchor text. Thus, the more often a mention is used to link to the same entity, the higher the Anchor-Link probability. This probability has been calculated on two different corpora. For the SPL dictionary this probability is based on the Wikipedia article corpus, and for the GCW dataset it is based on all web documents (see Section 10.5). The results of this experiment can be considered as an indicator for the degree of difficulty of the applied benchmark in terms of WSD. By simply using a popularity measure, a high recall and precision indicates a less difficult benchmark dataset. If a benchmark contains less popular entities, the disambiguation process can be considered more difficult.

10.6.2 *Discussion of Experiment Results*

For every experiment, a results table is given. The tables show the results for the four different dictionaries, represented by the columns, on the five different benchmarks, represented by the rows. For easy comparison, the number of dictionary entries and the number of distinct mentions for all benchmarks are provided along with their annotated entities. For all results, the total numbers as well as proportionally averaged values are given. This facilitates the comparison of benchmarks and dictionaries that are significantly different in size and number of annotations.

The experiments *mapping coverage*, *entity candidate count*, *maximum recall* and *recall and precision based on page link popularity* have been performed also using case-insensitive mentions and labels in the four different dictionaries. For comparison, these results are presented in the same tables of the respective experiments as the results of the case-sensitive experiments. Recall and precision based on Anchor-Link-Probability have not been calculated as the probabilities for case-insensitive anchors are not available within the SPL and GCW datasets.

MAPPING COVERAGE

- GCW achieves the highest coverage (between 94.67% and 100%) due to its employment of the largest dictionary containing 378 million entries and its construction method of anchor texts and linked Wikipedia articles in web documents.
- RDM performs the worst with only 25.19% on the Spotlight benchmark due to the lack of preprocessing; all labels are given with initial capital letters which is not common in the English language except for persons, places and organizations.
- Coverage for RDM increased by 69% (to 94%) when mentions in the Spotlight benchmark are looked up in the dictionary as case-insensitive entries. Also, for the Wikilinks benchmark, the coverage using the RDM dictionary increased by 16% to 76%. The increase of coverage is significant for the video dataset (throughout metadata of all sources) and the tag dataset – it amounts to up to 95% for metadata of user-generated tags within the video dataset. For the tag dataset, the mapping coverage is increased by 80% to 96%. This also reflects the specific characteristic of tags being mostly case-insensitive.
- The RDM dictionary consists of mainly case-sensitive labels (as no pre-processing has been performed). Persons, organizations and places are written with an initial capital letter in English language texts. Mentions of entities of those types are found in a case-sensitive dictionary such as RDM. In contrast, mentions of entities that are not of the types person, organization or place, such as *internet*, are not found in this dictionary. If a benchmark contains mainly mentions of entities of the types person, organization or place, the RDM dictionary achieves a high mapping coverage, as is the case for the KORE 50 benchmark. Case-insensitive selection must increase the coverage, especially if the benchmark contains entity mentions that are not of the types person, organization or place. This assumption is consolidated by the increased mapping coverage for the Spotlight and Wikilinks benchmarks and the type information of the mentioned entities in the benchmarks presented in Table 12.

Table 14: Coverage of mapped mentions – total count and percentage

BM \ Dictionary		SPL 2M entries		RDM 10M entries		AIDA 18M entries		GCW 378M entries		Mention Count
Spotlight		235	89%	65	25%	227	86%	258	97%	265
KORE 50		117	90%	129	99%	128	98%	130	100%	130
Wikilinks		107,669	56%	114,443	60%	115,646	60%	170,765	89%	192,008
Video Dataset	Authoritative	280	85%	102	31%	275	83%	314	95%	330
	Tags	238	72%	3	1%	222	67%	316	96%	329
	OCR	98	72%	71	52%	118	86%	127	93%	137
	ASR	1,608	89%	251	14%	1,573	87%	1,781	98%	1,812
Tag Dataset		269	53%	11	2%	264	52%	467	93%	505
Experiment with case-insensitive mentions and dictionary labels										
Spotlight		241	91%	249	94%	235	89%	258	97%	265
KORE 50		121	93%	130	100%	130	100%	130	100%	130
Wikilinks		114,278	60%	145,241	76%	128,139	67%	171,941	90%	192,008
Video Dataset	Authoritative	287	87%	310	94%	306	93%	317	96%	330
	Tags	296	90%	315	96%	307	93%	319	97%	329
	OCR	117	85%	132	96%	127	93%	134	98%	137
	ASR	1,651	91%	1,748	96%	1,679	93%	1784	98%	1,812
Tag Dataset		374	74%	427	85%	402	80%	487	97%	505

- Overall, the dictionaries perform very well, or even best, on the benchmarks that have been constructed for the evaluation of their respective applications: SPL – Spotlight, AIDA – KORE 50, and GCW – Wikilinks.
- Although the best coverage is achieved by GCW, RDM is close behind. Due to the size of GCW of 378 million entries, the performance of an algorithm suffers from queries on this dictionary with respect to time complexity (see Section 9.4). RDM only contains 2.6% of the entries of GCW. Therefore, look-ups on this dictionary are less time-consuming, resulting in only a short loss of recall.

The overall results are depicted in Table 14.

ENTITY CANDIDATE COUNT

- The KORE 50 benchmark is intended to contain mentions that are difficult to disambiguate. Overall, all dictionaries achieve highest entity count for this benchmark which entails high ambiguity.
- For the Wikilinks benchmark, all dictionaries achieve low entity candidate counts which shows that real world annotations do not seem to be difficult to disambiguate.
- The AIDA dictionary assigns most entity candidates on the KORE 50 benchmark, as this dictionary is constructed for evaluation on that benchmark and is supposedly enriched by labels especially for that purpose.
- KORE 50 contains many persons that are mentioned by their first name only. This results in a large number of entity candidates.
- The Wikilinks benchmark is annotated very sparsely and only those entities assumed to be "important" are linked.
- GCW achieves an extremely high number of entity candidates for the tags of the video dataset. This results from the fact that the term *wikipedia* occurs as an annotated term in the dataset as a user-generated tag and GCW lists 2,126,672 entity candidates for this term. Section 10.6.3 presents more results on the level of ambiguity of the different dictionaries under consideration.

Overall results are shown in Table 15.

MAXIMUM RECALL

- SPL and RDM do not contain all first names of persons as required for the benchmark KORE 50. Thus, the maximum recall decreases compared to mapping coverage.

Table 15: Amount of entity candidates for all mapped mentions – overall and averaged per mapped mention

BM \ Dictionary		SPL 2M entries		RDM 10M entries		AIDA 18M entries		GCW 378M entries		Mention Count
Spotlight		1,849	7.9	1,024	15.8	6,487	28.6	134,493	521.3	265
KORE 50		2,980	25.5	16,936	131.3	74,967	585.7	36,772	282.9	130
Wikilinks		188,748	1.8	244,977	2.1	299,193	2.6	1,346,446	7.9	192,008
Video Dataset	Authoritative	1739	6.2	2393	23.5	32000	116.4	141411	450.4	330
	Tags	906	3.8	10	3.3	2,446	11.0	2,168,061	6,861.0	329
	OCR	1,065	10.9	1,239	17.5	7,995	67.8	66,762	525.7	137
	ASR	8,524	5.3	6,124	24.4	26,791	17.0	390,850	219.5	1,812
Tag Dataset		1,036	3.8	42	3.8	4,568	17.3	85,718	183.5	505
Experiment with case-insensitive mentions and dictionary labels										
Spotlight		3,400	14.1	6,508	26.1	13,336	56.7	367,698	1425.2	265
KORE 50		3,079	25.4	16,946	130.4	75,326	579.4	46,244	355.7	130
Wikilinks		207,181	1.8	145,241	2.1	352,107	2.7	1.8 Mio.	10.6	192,008
Video Dataset	Authoritative	3,561	12.4	6,344	20.5	35,005	117.7	572,381	1,805.6	330
	Tags	4,034	13.6	8,213	26.1	10,747	35.0	2,130,016	6,677.2	329
	OCR	1,749	14.9	2,388	18.1	8,993	70.8	91,393	682.0	137
	ASR	23,430	14.2	39,962	22.9	56,947	33.9	1,231,063	690.1	1,812
Tag Dataset		4,389	11.7	7,906	18.5	21,187	52.7	199,377	409.4	505

Table 16: Maximum achievable recall – coverage of annotated entities (in the benchmark) for mentions contained in the list of candidates

BM \ Dictionary		SPL 2M entries		RDM 10M entries		AIDA 18M entries		GCW 378M entries		Mention Count
Spotlight		223	84%	60	23%	63	24%	241	91%	265
KORE 50		87	67%	93	72%	112	86%	110	85%	130
Wikilinks		82,338	43%	86,555	45%	82,565	43%	129,449	67%	192,008
Video Dataset	Authoritative	246	75%	90	27%	89	27%	294	89%	330
	Tags	204	62%	3	1%	27	8%	290	88%	329
	OCR	90	66%	68	50%	38	28%	111	81%	137
	ASR	1,433	79%	219	12%	324	18%	1635	90%	1,812
Tag Dataset		224	44%	10	2%	41	8%	409	81%	505
Experiment with case-insensitive mentions and dictionary labels										
Spotlight		224	85%	228	86%	75	28%	242	91%	265
KORE 50		89	68%	93	72%	112	86%	110	85%	130
Wikilinks		86,955	45%	106,713	56%	92,824	48%	130,335	68%	192,008
Video Dataset	Authoritative	253	77%	263	80%	113	34%	298	90%	330
	Tags	254	77%	277	84%	102	31%	298	91%	329
	OCR	109	80%	120	88%	43	31%	125	91%	137
	ASR	1,452	80%	1,523	84%	388	21%	1,643	91%	1,812
Tag Dataset		316	38%	348	69%	162	32%	454	90%	505

- AIDA performs poorly on the Spotlight benchmark due to the structure of this dictionary. It contains a large number of persons' first names. Apparently, the dictionary does not reflect labels for entities in manually annotated texts.
- For RDM, the maximum recall increases by 10% to 63% respectively for the two benchmarks Wikilinks and Spotlight, if mentions are looked up as case-insensitive. This is a reflection of the structure of the benchmarks and the increased coverage of mapped mentions.
- For the Wikilinks benchmark, the maximum possible recall is low compared to the other two benchmarks. This results from the fact that this benchmark cannot be considered as a gold standard (see Section 10.3). If a mention is annotated with a wrong entity, there is a high probability that this entity is not contained in the list of entity candidates.
- GCW and RDM achieve the highest possible recall on video dataset and tag dataset in comparison to SPL and AIDA. On the contrary, the decrease in the maximum recall compared to mapping coverage is lowest for RDM and GCW (between 6% and 14%) in comparison to SPL and AIDA. AIDA achieves a mapping coverage of approximately 93% on the video dataset across metadata of all sources. However, the maximum recall drops to under 35% for metadata of authoritative sources, user-generated tags and OCR source and as low as 21% for metadata of ASR sources.

Overall results are shown in Table 16.

RECALL AND PRECISION ACHIEVED BY POPULARITY – INCOMING WIKIPEDIA PAGE LINKS OF ENTITY CANDIDATES

- Notably, GCW performs poorly on all benchmarks compared to the maximum possible recall due to a high entity candidate count. Apparently, entity candidate lists often contain more popular but incorrect entities.
- In the KORE 50 benchmark, due to many annotated first names, entity candidate lists contain numerous prospective entities. Apparently, the correct candidate is often not the most popular one compared to the other candidates. This explains the poor performance of all dictionaries on the KORE 50 using page link popularity.
- Compared to the maximum possible recall (of all dictionaries) on the KORE 50, the achieved recall is very low using a popularity measure in a simplified disambiguation process. This con-

firms the intention of the benchmark to contain mentions that are hard to disambiguate.

- For the case-sensitive experiment, RDM achieves a relatively low recall and high precision for all benchmarks, especially for the video dataset and the tag dataset. Precision and recall for the other three dictionaries are mostly balanced on a low level. For the case-insensitive experiment RDM also shows a balanced ratio of recall and precision on a mid-ranged level.

Case-sensitive results of all datasets are shown in Table 28; case-insensitive results are shown in Table 29 in the appendix.

RECALL AND PRECISION ACHIEVED BY POPULARITY – ANCHOR--LINK PROBABILITY IN WEB DOCUMENT CORPUS (GOOGLE POPULARITY)

- In general, this popularity based on mentions and mapped entity performs better than popularity based only on the entities' incoming Wikipedia page links.
- The recall of GCW dictionary is increased between 13% and 55%. The increases of the recall for the RDM and AIDA dictionaries are not significant compared to page link popularity.
- As for the case-sensitive experiment using Wikipedia page links as indicators of popularity, RDM achieves low recall and high precision over all benchmarks. As previously stated, this results from the low mapping coverage for this dictionary.
- GCW achieves a relatively balanced ratio of recall and precision on a mid-ranged to high level over all benchmarks.

RECALL AND PRECISION ACHIEVED BY POPULARITY – ANCHOR-LINK PROBABILITY IN WIKIPEDIA CORPUS

- For the Spotlight and Wikilinks benchmarks, this popularity measure achieves higher recall and precision than the popularity measure provided by GCW dictionary. It is likely that this results from the fact that the Wikipedia corpus is composed by experienced authors and linked texts are considered to be reliable.
- Again, RDM achieves low recall and high precision over all benchmarks. This results from the fact that the mapping has been performed sensitive to case.
- Although the probability for this experiment originates from the SPL dictionary, the differences between GCW and SPL are not remarkable. Both dictionaries provide their own probabilities,

and the probabilities perform best when applying the respective dictionary. However, as seen in Table 30, the differences in terms of recall and precision are not exceptional.

Overall results are shown in Table 31 in Appendix A.

GENERAL FINDINGS

- For a simplified disambiguation process, the Anchor-Link popularity performs better than popularity based on Wikipedia page links. Anchor-Link popularity calculated on the Wikipedia corpus performs better than the measure calculated by Google on the web document corpus.
- Dictionaries perform best on the benchmark constructed for the evaluation of the dictionaries' applications.
- Compared to the maximum possible recall (of all dictionaries) on the KORE 50 benchmark, the achieved recall is very low using a popularity measure as a simplified disambiguation process. This confirms the intention of the benchmark to contain mentions that are difficult to disambiguate.
- SPL performs very well over all benchmarks, especially using its popularity measure. Taking into account its size (2.2 million entries) compared to the GCW dictionary (378 million entries), this is a surprising discovery. However, the case-insensitive experiments show even better results for RDM in terms of mapping coverage and maximum possible recall.
- The SPL popularity measure has been calculated based on the linked Wikipedia articles within the Wikipedia article corpus. Most of the Wikipedia articles have been composed by experienced authors who know how to write and distribute links within the corpus. This could be an explanation for why the Wikipedia based Anchor-Link probability performs better than the popularity based on web documents.
- The best mapping coverage is achieved by GCW, with RDM close behind. Look-ups in GCW are very time-consuming due to the size of the dictionary. Look-ups in RDM are less time-consuming, which results in only a short loss of recall. Also, on average, the amount of entity candidates is 3.6% for RDM in comparison to the GCW. A high number of entity candidates results in a very time-consuming disambiguation process in the case of the application of the link graph analysis as described in Section 8.2.4. Therefore, RDM constitutes a good trade-off between time-consumption for the overall algorithm, mapping coverage and achievable recall.

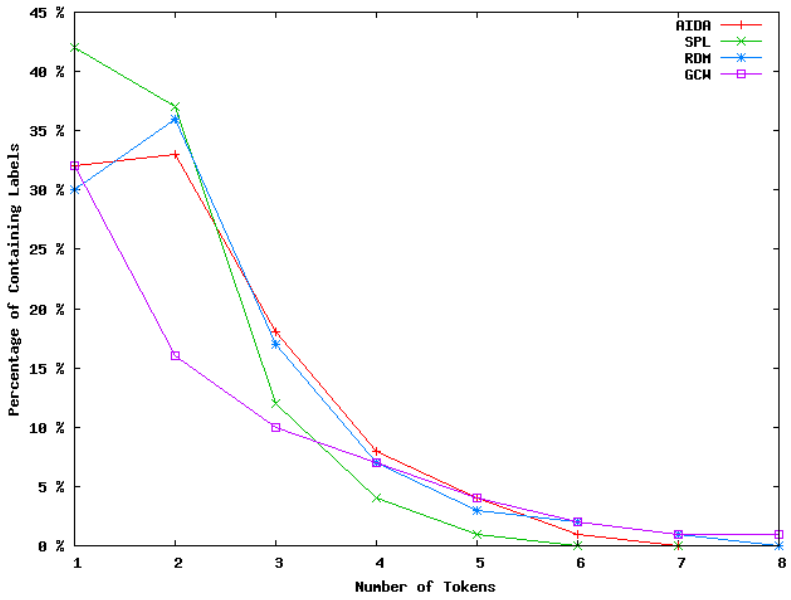


Figure 21: Distribution of the amount of tokens for labels contained in the four dictionaries.

10.6.3 *Ambiguity of Dictionaries*

Interesting insights about the dictionaries come from the list of most ambiguous terms within the dictionaries. Table 17 shows the top ten ambiguous terms for every dictionary. The lists support the previously mentioned assumptions about the dictionaries:

- AIDA contains a large number of personal first names. As shown, the labels are case-sensitive as well as provided in upper case. It is unclear is why they are not given in lower case.
- The anchors gathered on websites for the GCW dictionary mostly refer to Wikipedia articles, but do not use the article label or a synonym. Nine of the top ten most ambiguous anchors contain the word *{W|w}ikipedia*.
- Whereas RDM and SPL contain very ambiguous personal family names, the level of ambiguity differs extremely between the two dictionaries.

10.6.4 *Number of Tokens of Containing Labels*

The dictionaries have been analyzed for the number of tokens of the containing labels. This analysis has been performed to identify the most common number of tokens. Section 8.2.1 introduced a tag preprocessing method for the combination of tags to form tag groups. The maximum group size of tags influences the complexity of the algorithm. The larger the group size, the more combinations must

Table 17: Top 10 of most ambiguous terms in the four different dictionaries.

SPL		RDM		AIDA		GCW	
Term	Count	Term	Count	Term	Count	Term	Count
Georges	413	Smith	4,067	JOHN	22,490	View original Wikipedia Article	3,436,360
O'Neill	307	Jones	2,578	John	22,488	Wikipedia	3,397,240
Murphy	228	Williams	2,452	WILLIAM	12,350	Wikipedia article	3,092,350
Davies	213	Brown	2,391	William	12,350	View on Wikipedia	3,016,494
Charles	211	Johnson	2,232	JAMES	9,760	original Wikipedia article	2,913,969
Wright	202	Wilson	1,742	James	9,760	View article on Wikipedia	2,832,115
Saint-Pierre	190	Lee	1,725	DAVID	9,278	here	2,502,884
Silva	187	Taylor	1,661	David	9,278	wikipedia	2,126,672
Cox	171	Miller	1,550	ROBERT	8,800	View article on Wikipedia »	2,087,894
Ellis	171	Anderson	1,496	Robert	8,800	Source:Wikipedia	1,824,735

be calculated. Therefore, the dictionaries have been analyzed for the most common numbers of tokens of the contained labels. As shown in Figure 21, SPL and GCW mostly contain labels consisting of one token (42% and 32%, respectively), followed by labels consisting of two tokens (37% and 16%, respectively). AIDA and RDM mostly consist of labels possessing two tokens (33% and 36%), closely followed by labels consisting of one token (32% and 30%). Overall, the percentage of labels possessing more than three tokens decreases significantly for all four dictionaries. Labels consisting of three or less tokens amount to 83% for AIDA and RDM and as much as 91% for SPL. For GCW, these labels amount to 58%.These results substantiate the decision for a maximum group size of three for the tag combination algorithm.

The label containing the most tokens consists of 164 tokens in AIDA and 264 tokens in GCW, which must be a bug resulting from the automatic extraction of the labels. The label with the highest number of tokens in RDM consists of 64 tokens and 28 tokens in SPL.

Additionally, the curves shown in Figure 21 reveal the similar characters of the dictionaries. The curves of RDM and AIDA resemble one another, as both dictionaries have been (at least partly) created from redirect and disambiguation labels of DBpedia. The curves of GCW and SPL also resemble each other, as both dictionaries have been created from anchor texts in web documents.

Basically, SPL reflects the structure of mentions of semantically annotated documents as required for a semantic search. Therefore, the distribution of the number of tokens over all labels is applied for the ranking of context items with respect to the number of tokens of their textual terms (see Section 7.1.2's *Number of Tokens*).

10.6.5 *Discussion*

The most important measures for the decision about a dictionary are the mapping coverage and the maximum achievable recall, because the dictionary should contain all important terms mentioned in the benchmark and the entity candidates should include the correct one for the type of benchmark. GCW achieves the highest results over all benchmarks for both measures, but also provides a high entity count which means the entity look-up takes more time because of its size with many candidates to be checked during the disambiguation process. RDM provides similar high coverage and maximum possible recall, but at the same time provides a reasonable amount of entity candidates. Therefore, RDM in its case-insensitive form has been applied for the following evaluations.

10.7 EVALUATION RESULTS

This section presents the evaluation results of the proposed algorithms. The tag processing approach introduced in Section 8.2.1 has been examined separately as it is a very specific task, using the dataset introduced in Section 10.3.2.

The disambiguation approach presented in Section 8.2 has been evaluated using the Spotlight benchmark (see Section 10.3.3) as this benchmark has been utilized for the analysis of DBpedia Spotlight and the comparison of their approach and the presented approach is easy to present. The results are presented in Section 10.7.3.

The context model and the semantic analysis process using the context model, as well as the negative context approach, have been evaluated on the video metadata benchmark (see Section 10.3.1). The results are presented in Sections 10.7.5 and 10.7.7. Although the context model has been developed for the specific purpose of the semantic analysis of video metadata, the model can be applied for simple continuous texts. Therefore, the approach has been evaluated on an independent dataset. The results are presented in Section 10.7.8.

For all evaluations, an RDM dictionary has been applied (see Section 10.5.4) . As described in Section 10.6.2, this dictionary constitutes a good trade-off between high time complexity for the overall algorithm, mapping coverage and maximum possible recall.

10.7.1 *Evaluation of Tag Processing*

The tag processing and disambiguation approach presented in Section 8.2.1's *Tag processing*, has been evaluated on the tag benchmark described in Section 10.3.2 and against the DBpedia Spotlight annotation service. This analysis step has been evaluated separately, because user-generated tags require special attention for the recognition of named entities and important terms within a group of tags. As shown in Table 18, the tag processing and disambiguation approach clearly outperforms annotation approaches customized for simple natural language texts. For both datasets, the presented approach to tag combination achieves higher recall and precision than Spotlight. Consequently, the presented approach is used in the semantic analysis process for the recognition of named entities and important terms in user-generated tags.

10.7.2 *Evaluation of the Detection of Named Entities in Continuous Text*

As previously stated in Section 10.1, the results of a semantic analysis process depend on the algorithm that detects the named entities within a continuous text. If the algorithm extracts inappropriate terms, the disambiguation algorithm has to work with incomplete

Table 18: Comparison of the presented tag processing and disambiguation approach against DBpedia Spotlight for tag benchmarks

	Segment Contexts			Timestamp Contexts		
	256 tags 300 entities assigned			315 tags 485 entities assigned		
	Recall	Precision	F_1	Recall	Precision	F_1
Presented Approach	78.0	64.0	69.0	81.0	41.0	54.0
Spotlight	39.0	34.0	36.0	42.0	39.0	40.0

or erronous information, resulting in poor results. The detection algorithm presented in Section 8.2.1 takes into account five different types of textual terms detected in a text (see Table 9):

- simple nouns
- proper nouns
- combined (proper) nouns
- adjectives followed by nouns
- (proper) nouns connected by a preposition

The POS tagger determines the type of each token in a continuous text. Errors resulting from the POS tagger are dragged into the detection process and thereby textual items might be incorrectly identified as prospective named entities or important terms. For this evaluation, the annotation of the benchmarks have been compared to the annotations of the algorithm's results taking into account not the actual annotated entity, but rather the position and length of the annotation. Recall, precision, F_1-measure and accuracy have been calculated for comparison. The true positives (TP) constitute the characters in the text that have been identified as part of a named entity by both the annotation of the benchmark and the detection algorithm. The false positives (FP) constitute the characters that have been identified by the detection algorithm, but are not annotated in the benchmark. The false negatives (FN) constitute the characters annotated in the text, but not detected by the algorithm as part of a prospective named entity. And the true negatives (TN) constitute the characters that have not been annotated in the benchmark and also have not been annotated by the algorithm. Results are shown in Table 19.

The results show that annotations made by human annotators differ from actual detected prospective named entities. The comparatively low precision is a result of the algorithm's *over-annotation*. Thus, a human annotator annotates fewer entities in a continuous text than

Table 19: Comparison of annotations in benchmarks and of prospective named entities detected by proposed algorithm.

	Recall	Precision	F_1-measure	Accuracy
Spotlight Benchmark	72.0	55.0	62.5	79.0
Video Benchmark	80.5	48.5	60.5	80.0

those detected by the algorithm. The text in the following sentence is underlined where a human annotator would mark an entity[5]:

Last year, I told you the story, in seven minutes, of Project Orion.

The next sentence shows the detection result of an algorithm, taking into account the word types listed in Table 9:

Last year, I told you the story, in seven minutes, of Project Orion.

These examples demonstrate the low precision of the algorithm's annotation result. The low recall might be explained by the error of the POS algorithm as well as by different combinations of word types compared to the benchmark. The accuracies for both benchmarks are fortunately high which reflects the high accuracy of the algorithm's annotated text. The presented semantic analysis strives to provide annotated documents for a semantic search. Therefore, important key terms are annotated in addition to named entities. These important key terms support the exploration of (video) documents in a further way than simply based on named entities. Therefore, the low precision in comparison to a human annotator is acceptable for the application of the presented algorithm.

10.7.3 *Evaluation of the Disambiguation Approach*

Section 10.7.1 provided results for the evaluation of the disambiguation approach introduced in Section 8.2, in combination with the tag processing method. However, the approach has also been evaluated using the Spotlight benchmark (see Section 10.3.3) against DBpedia Spotlight as individual process on continuous text. As presented in [75], DBpedia Spotlight achieves an F_1-measure of 45.2% when no further configuration is applied to the annotation service.

The presented disambiguation approach uses continuous text as input. The context model and the ordering of context items have not been applied for this evaluation. The approach achieves a recall of 53% and a precision of 40%, resulting in an F_1-measure of 45.5%. Thus, it is shown that the disambiguation algorithm – as a separate algorithm – is competitive with the state-of-the-art annotation service DBpedia Spotlight and even performs slightly better.

5 The example is taken from the video benchmark.

As shown in Tables 30 and 31, a maximum recall of 75% and a precision of 85% are achieved by the simple disambiguation process using probability scores. These results are generated by mapping the annotated terms of the respective benchmarks and the analysis of the assigned entity candidates. Therefore, these results can only be achieved when the exact terms (as annotated in the ground truth) are detected by the applied algorithm. The difference between the achieved results of the proposed algorithm (but also DBpedia Spotlight as a state-of-the-art annotation service) and the maximum possible results (shown in Tables 31 and 30) reveal the difficulty of the actual spotting and recognition algorithm.

The evaluation using the proposed context model on the Spotlight benchmark is described in Section 10.7.8.

10.7.4 *Evaluation of Score Weights*

The analysis methods of the disambiguation process presented in Sections 8.2.3 through 8.2.6 generate a score within the range [0.0...1.0]. The final score of the disambiguation process is calculated from these individual scores, where some methods might be more important for the correct context-dependent interpretation than others. Therefore, the scores might be weighted for the calculation of the final score. An empirical study has been performed on the DBpedia Spotlight dataset (see Section 10.3.3 for a description of the dataset). For this study, the co-occurrence analysis, the three different link graph analyses, the coreference analysis and the heuristic based on the matching of the main label of the candidate and the term have been utilized. The scores have been weighted with values in a range of [0.0...1.0] in increments of 0.2. This increment is considered a trade-off between the high accuracy of the determined weights and the costs for the determination. A total of 46,656 runs have been performed. The study has aimed to detect the weight assignments for the single scores achieving the highest recall and precision. As a result, instead of a single combination, 801 combinations of weight assignments achieve the highest F_1-measure. Therefore, a specific recommendation for assigned weights cannot be suggested. But, several tendencies have been identified:

- All scores achieved by link analysis methods should be weighted higher than the other scores.
- A score achieved by coreference analysis should be weighted higher than entity-based popularity measures.
- Co-occurrence is more crucial than any entity-based popularity analysis, because it is a context-dependent measure. However, it is less crucial than the coreference analysis.

Thus, a ranking according to the importance of the analysis methods can be derived based on these assumptions:

1. link graph analyses
2. coreference analysis
3. co-occurrence analysis
4. main label matching heuristic

The results show that the context-dependent analyses are more important than the context-independent analyses (main label matching). Obviously, links between entities represent more important relationships than descriptive texts. This means that descriptive texts might contain much information about an entity which overlaps with information about other entities. Meanwhile, links between entities reveal semantic relationships and are a strong indicator for a disambiguation task. The coreference analysis also reveals semantic relationships within the context. If the same entity is mentioned in a context by different terms, this might be a strong indicator that this entity is the correct interpretation for both terms.

As a consequence, the weights for the separate analyses have been determined to concrete values following the previously introduced assumptions:

- $w = 0.8$ for all link graph analyses
- $w = 0.6$ for the coreference analysis
- $w = 0.4$ for the co-occurrence analysis
- $w = 0.2$ for the main label matching heuristic

The weights are distributed linearly according to the ranking of the analyses. The order of the weights is considered more important than the actual determination of the concrete values for the highest and the lowest weights. More precise weights might be examined in future work.

The relevance of the separate analysis methods strongly depend on the underlying knowledge base. These evaluations have been performed on the basis of the DBpedia. As stated in Section 8.1.3, the DBpedia contains crucial information about the entities: descriptive texts (required for the co-occurrence analysis) as well as the semantic relationships (required for the link graph analyses). Therefore, the determined weights can only be applied for knowledge bases structured similarly to DBpedia. Future work might focus on a general determination of the score weights or a definition of rules according to the characteristics of the underlying knowledge base.

Table 20: Compared results of *conTagger* and the simple segment-based approach including the significance measure with respect to the difference of the approaches (Θ based on F_1-measure of both approaches).

	conTagger			*Simple Approach*			Θ
	R	P	F_1	R	P	F_1	
Authoritative	**60.0**	**54.5**	**57.0**	52.0	46.0	49.0	0.1553
Tags	**71.0**	**69.5**	**70.0**	61.0	60.0	60.5	0.0023
ASR	55.0	**61.0**	**58.0**	56.5	38.0	45.5	0.0036
OCR	**56.0**	**24.0**	**34.0**	44.0	17.5	25.0	0.0017
Segments	54.0	**58.0**	**56.0**	57.0	39.0	46.5	0.0050
Videos	56.0	**48.0**	**52.0**	57.0	30.0	39.5	0.0340

10.7.5 *Evaluation of the Context Model*

The *conTagger* has been evaluated on the video benchmark introduced in Section 10.3.1.

COMPARISON TO SIMPLE SEGMENT-BASED APPROACH As a first step, the *conTagger* has been compared to a simple segment-based approach using a random order of the context items, but the same disambiguation approach as the *conTagger*. The results of both approaches are shown in Table 20. The results are aggregated according to different sources and different relevance views. For the different sources, recall and precision are calculated per segment and averaged over all segments of all five videos. For segments, recall and precision are calculated for every segment over all sources and averaged over all segments for all videos. The evaluation results for videos are calculated respectively. As shown in Table 20, recall, precision and F_1-measures are increased throughout all sources and also for aggregated results for segments and videos. Most notably, the *conTagger* achieves significantly good results on the metadata items with lower confidence values, as OCR and ASR results. The overall evaluation of annotated entities per segment and video confirms the significant results.

STATISTICAL HYPOTHESIS TEST Additionally, a statistical hypothesis test has been performed. The hypothesis tests demonstrates that *conTagger* is different than the simple segment-based approach and that the results have not been achieved simply by chance. A significance level Θ is calculated for the test. A value of $\Theta \leqslant 0.05$ is considered *low enough* to prove the difference of the systems [80]. Θ has been calculated using *approximate randomization* as follows [84]:

1. Two approaches A and B are provided with the same input data.
2. Both approaches return a set of results $O_A = \{o_A^1, ..., o_A^n\}$ and $O_B = \{o_B^1, ..., o_B^m\}$. Calculate $t(O_A, O_B)$.
3. Taking the results of both approaches, a new set is created: $Z = \{o_A^1, ..., o_A^n, o_B^1, ..., o_B^m\}$.
4. Two new sets O'_A and O'_B are created by sampling with replacement from set Z, with $|O'_A| = |O_A|$ and $|O'_B| = |O_B|$. Calculate $t(O'_A, O'_B)$.
5. Repeat Step 4 R times. For all times where $t(O'_A, O'_B) \geqslant t(O_A, O_B)$ increase a counter r.
6. Calculate $\Theta = \frac{r+1}{R+1}$.

For this evaluation, $R = 1000$ and the function t calculated for comparing the result sets is based on the difference of the F_1-measure of both sets:

$$t(O_A, O_B) = |F_1(O_A) - F_1(O_B)|$$

Θ has been calculated for all source types separately and aggregated for all segments. The overall Θ for the video benchmark has been calculated as the *harmonic mean* of all θ_i of all k video segments of the benchmark:

$$\Theta = \frac{1}{k} \cdot \sum_{i=1}^{k} \frac{1}{\theta_i}$$

Likewise, Θ for segments and videos has been calculated for aggregated segments respectively videos regardless of the source type.

As shown in Table 20, Θ is very low ($\leqslant 0.05$) for the sources *Tags*, *ASR* and *OCR*. This shows the significant difference of the approaches for metadata items of these source types. This reflects the intention of the context model: first and foremost, the context model has been developed to enable improved analysis results for metadata items with a low confidence value. Metadata items from authoritative sources possess a high confidence value by nature, because of the high reliability assigned to the source type. Therefore, the application of the context model shows the most significant results on the metadata items of source types *Tags*, *ASR* and *OCR* and the significance level for authoritative sources is comparatively high ($\Theta = 0.1553$). The significance level for the results aggregated over the entire video is slightly increased compared to the separate significances. Yet, this aggregated significance level is an imprecise measure, because the actual time and segment assignment of the occurrence and the origins of the results are not considered.

The overall results support Hypothesis 9.1.1, that the results of the disambiguation process are improved if context items possessing a high confidence value are disambiguated first in a heterogenous context and serve as reference points for subsequent disambiguations.

COMPARISON TO STATE-OF-THE-ART TOOLS The results of *conTagger* also have been compared to results of the following state-of-the-art annotation tools[6] using the video benchmark:

- DBpedia Spotlight
- Wiki Machine
- TagMe
- AIDA

For the application of these services, all metadata items assigned to a video segment – constituting a context – have been processed as one continuous text. The sources of the textual information have been identified by tracking the position within the input text. This enables an evaluation of all services divided by sources.

PARAMETER SETTINGS

- DBpedia Spotlight: Spotlight has been used by the employment of the web service. Two parameters can be set: confidence and support. For the presented evaluation the confidence has been set to 0.2 and support has been set to 20. These are the default settings as suggested by the Spotlight developers.
- Wiki Machine: The annotations by the Wiki Machine have been performed by the developers of the tool. The output of the tool includes a disambiguation score (similar to the disambiguation score presented in this work). All annotations that achieved a score higher than 0.0 have been added to the final result for the evaluation.
- TagMe: This service has been used by the employment of the web service. The annotated result comprises a score called `rho`. Following the suggestion of the TagMe developers, an annotated entity has been added to the result set only if `rho` exceeds the value 0.1.
- AIDA: For the evaluation of AIDA on the video benchmark the engine has been build using the sources provided on gitHub[7] and the entity repository has been set up. The output of the

6 See Section 4.3 for further details on the services.
7 `https://github.com/yago-naga/aida`

algorithm includes a disambiguation score. This score must exceed the value of 0.1 (as suggested by the AIDA developers) for the annotation to be added to the result.

The evaluation results according to the different sources as well as video and segment-based evaluations are depicted in Table 21.

The comparison of the results shows that *conTagger* outperforms the other state-of-the-art tools on almost all measures and for all sources, especially on metadata of sources with low confidence values. For better comparison, the 95% confidence intervals based on bootstrap percentiles have been calculated. Figure 22 shows the visual overview of the compared results and the 95% confidence intervals of F_1-measure, recall and precision separately for all source types and aggregated over segments and videos. The confidence intervals have been determined following the percentile bootstrap method introduced by Efron and Tibshirani [28]. The confidence intervals have been calculated separately for recall, precision and F_1-measures. The intervals have been calculated using $n = 1000$ bootstrapped sample sets from the original samples. The intervals have been calculated as follows:

1. Calculate mean $\overline{x}$ of the original samples.
2. Create bootstrap by resampling with replacement from the original sample. Calculate mean of the bootstrapped sample $\overline{x}_i^b$.
3. Repeat Step 2 n times.
4. Calculate sample variance of all bootstrapped means: $s^2 = \sum_{i=1}^{n} (\overline{x}_i^b - \overline{x})^2$
5. Calculate standard deviation: $\sigma = \frac{s}{\sqrt{k}}$, where $k = 6$ denotes the size of the original sample.
6. Calculate confidence interval with α-percentile $t_{0.95,6}$=1.9432 for $\alpha = 0.95$ and sample size $k = 6$:
   $CI_{lower} = \overline{x} - (1.9432 * \sigma)$
   $CI_{upper} = \overline{x} + (1.9432 * \sigma)$

As shown in Table 21 and Figure 22, the results for the *conTagger* mostly range above the upper bound of the confidence intervals. These results support the assumption of Hypothesis 9.1.1 and confirm the contribution achieved by the application of the proposed context model.

EVALUATION OF HYPOTHESIS 9.1.2 For evaluating Hypothesis 9.1.2[8], the context items have been disambiguated using the entire video as context as well as for only context items of the same video segments for comparison. Evaluation results are shown in Table 22.

8 *The context of context items within a video document should be restricted to segments of coherent content.*

Table 21: Evaluation results of *conTagger* compared to simple approach, DBpedia Spotlight, Wiki Machine, TagMe and AIDA (R = Recall, P = Precision, F_1 = F_1-Measure).
CI_{lower} and CI_{upper} represent the lower and upper bounds of the 95% confidence intervals based on bootstrap percentiles (using $n = 1000$ bootstraps of the original samples).

		CI_{lower}	CI_{upper}	conTagger	Simple Approach	Wiki Machine	Spotlight	TagMe	AIDA
Authorative	R	40.0	53.0	**60.0**	57.0	59.5	50.0	48.5	8.0
	P	48.5	54.0	54.5	46.0	56.5	44.0	**62.0**	43.0
	F_1	41.0	52.0	52.0	49.0	**58.0**	47.0	54.5	13.0
Tags	R	39.0	55.5	**71.0**	61.0	44.0	60.0	42.0	2.0
	P	57.0	63.0	**69.5**	60.0	62.0	59.0	66.5	41.5
	F_1	42.5	57.5	**70.0**	60.5	51.5	59.5	51.5	3.5
ASR	R	39.5	53.0	55.0	56.5	**61.5**	56.0	41.5	6.0
	P	43.0	49.0	**61.0**	38.0	50.0	34.0	45.5	46.5
	F_1	37.5	48.5	**58.0**	45.5	55.0	42.5	43.5	11.0
OCR	R	31.5	42.0	**56.0**	44.0	24.5	47.0	39.0	9.5
	P	18.5	20.5	**24.0**	17.5	18.0	18.0	22.5	16.0
	F_1	22.0	27.0	**34.0**	25.0	21.0	26.0	28.5	12.0
Segments	R	39.0	52.0	54.0	57.0	57.0	**59.0**	41.0	5.5
	P	42.0	47.0	**58.0**	39.0	49.0	35.0	45.0	40.0
	F_1	37.0	47.5	**56.0**	46.5	52.5	43.5	43.0	9.5
Video	R	39.5	44.0	56.0	57.0	**58.0**	54.0	43.0	7.0
	P	37.5	42.5	**48.0**	30.0	43.0	31.0	43.0	44.5
	F_1	35.0	44.0	**52.0**	39.5	49.5	39.5	43.0	12.0

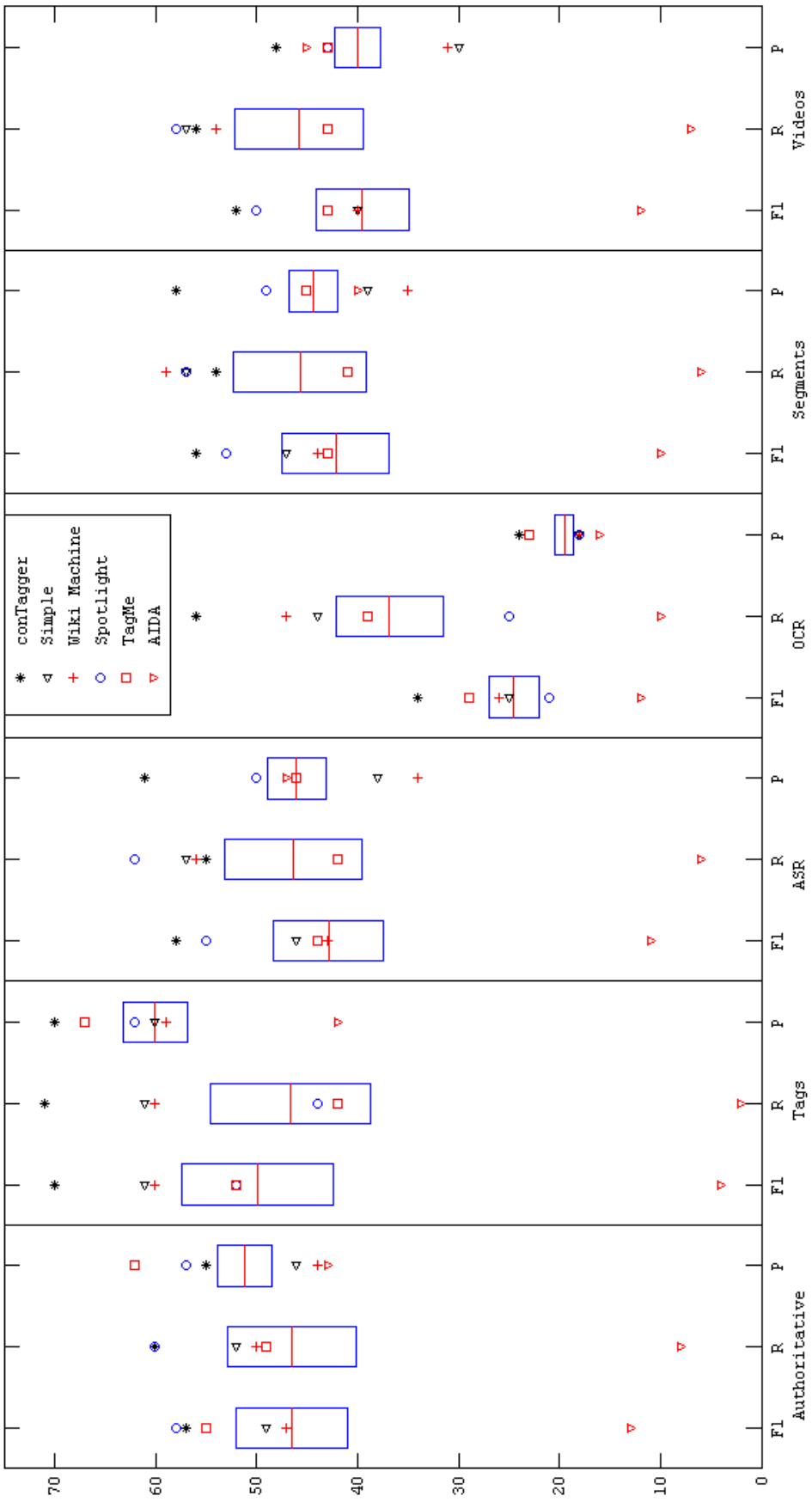


Figure 22: Results of *conTagger* compared to the simple approach, DBpedia Spotlight, Wiki Machine, AIDA and TagMe.

Table 22: Evaluation of Hypothesis 9.1.2. R = Recall, P = Precision

	ASR		OCR		Tags	
	R	P	R	P	R	P
conTagger, Segment-Based	**55.0**	**61.0**	**56.0**	**24.0**	**71.0**	**69.5**
conTagger, Video-Based	53.0	46.0	51.0	21.0	69.0	68.0

As anticipated, the disambiguation results are improved using content-based segments as context. The results for ASR metadata items are especially different in recall and precision for both versions. This probably follows from the fact that in the video dataset there are many more ASR metadata items, and because speech usually consists of more wide-spread content in terms of context information. However, recall and precision are not significantly different, which results from the homogenous character of the single videos of the dataset and their video segments.

LEVEL OF AMBIGUITY: NUMBER OF CANDIDATES VS. ASSIGNED CLASS As described in Section 7.1, the ambiguity of a context item can be defined also by the number of entity candidates. Therefore, the disambiguation process has been evaluated using the inverted normalized number of entity candidates instead of the class cardinality measure. Better evaluation results were achieved by using the class cardinality (as proposed in Section 7.1). F_1-measures for all source types were lower by an average of 5% when using the ambiguity measure based on the number of entity candidates. Obviously, a low number of entity candidates does not necessarily mean that the correct entity is amongst the few candidates. Therefore, the ambiguity measure is set according to class cardinality of an assigned class.

PARAMETER OPTIMIZATION FOR DYNAMIC CONTEXT CREATION During the disambiguation process, three parameters are available for the dynamic context creation, wherein select which context items are added to a context to disambiguate another context item:

- relevance with respect to context boundaries,
- the confidence value, and
- the disambiguation score (only applicable for previously disambiguated context items).

RELEVANCE Authoritative context items are disambiguated using other authoritative context items. In this case, the context boundaries are determined by the source type. Time-referenced metadata, such

Table 23: Best parameter configurations for the dynamic context creation divided in source types.

Source	Confidence Value	Disambiguation Score
ASR	0.7	0.2
OCR	0.7	0.2
Authoritative	0.25	0.0
Tags	0.4	0.0

as context items derived from OCR, ASR or tags, are not used as context items for the disambiguation of authoritative context items. Context boundaries for context items derived from ASR, OCR and user-generated tags are defined by segments of coherent content. Context items of different segments do not belong to the same context. In addition, authoritative context items are used for the disambiguation of context items from OCR, ASR and user-generated tags.

CONFIDENCE VALUE AND DISAMBIGUATION SCORE Those two parameters can be used to decide whether or not a context item is added to a context for disambiguation. Exhaustive test runs have been processed to determine the best suited thresholds for the confidence value and the disambiguation score with respect to the dynamic context creation. The values for both parameters range between 0 and 1. Therefore, the context creation and subsequent disambiguation process has been performed with all combinations of confidence values and disambiguation scores increasing the parameters in increments of 0.05, resulting in 441 runs. Subsequently, the parameters settings which achieve the best recall and precision aggregated over different source types have been identified.

The best recall and precision results for context items which originate from OCR and ASR algorithms (featuring the lowest confidence values) are achieved by creating the context from context items with a minimum confidence value of $c = 0.7$.

Authoritative context items are disambiguated only using authoritative context items as context (see Section 9.1.2). Therefore, they are disambiguated based on context items with highest confidence values in any case, because authoritative context items achieve the highest confidence values compared to context items derived from non-authoritative and non-human sources. The lowest possible confidence value for a context item from authoritative sources is calculated with $c = 0.4525$. The identified minimum threshold for the confidence value when adding a context item to a context is comparatively low, with $c = 0.25$.

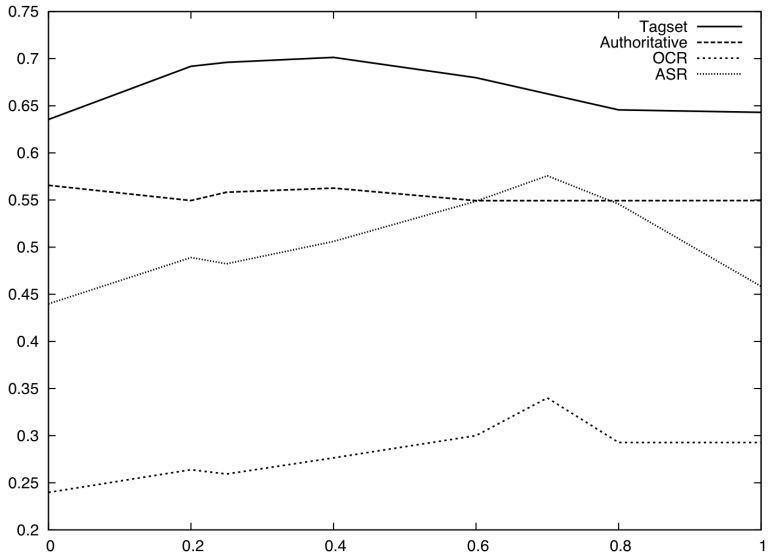


Figure 23: Achieved F_1-measures for different confidence values as threshold for dynamic context creation.

The minimum threshold for the disambiguation of user-generated tags is determined to be mid-range with $c = 0.4$. This means that for the disambiguation, some other time-referenced context items (from OCR or ASR analysis) are used as context items, but they are not all used, as the lowest possible confidence value for time-referenced metadata is calculated with $c = 0.285$.

The threshold for the disambiguation score when adding context items to a context is $s = 0.2$ for context items from OCR and ASR and $s = 0.0$ for authoritative context items and tags. Apparently, the achieved disambiguation score is not as important as the confidence value for the context items used as influencing items in the disambiguation process.

The parameter settings which achieve the best analysis results are shown in Table 23. The development of the F_1-measure for different parameter settings are shown in Figure 23 and Figure 24, where the x-axis denotes the value of the evaluated parameter (confidence value or disambiguation score) and the y-axis represents the achieved F_1-measure.

The achieved evaluation results support the premise that the characteristics and the use of contextual factors of different metadata items support the semantic analysis process.

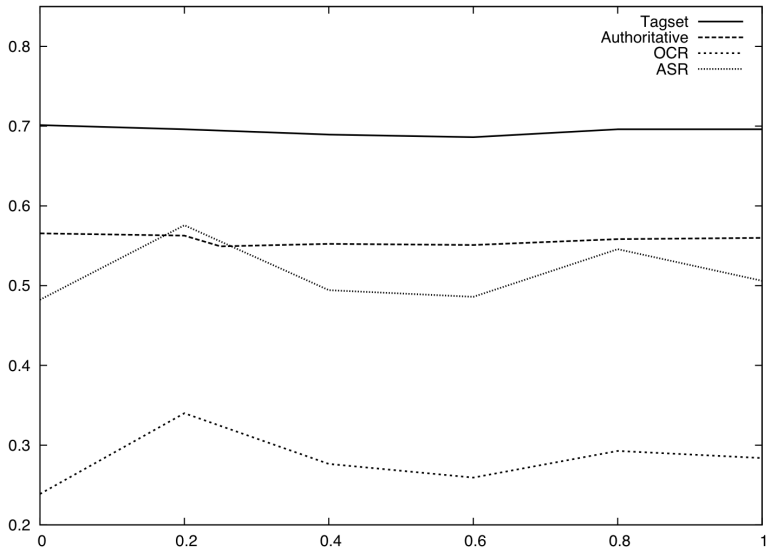


Figure 24: Achieved F_1-measures for different disambiguation scores as threshold for dynamic context creation.

10.7.6 *Evaluation of Influence of Constituents of Contextual Description*

The evaluation results presented above have been achieved by calculating the confidence value from all five constituents: *source reliability*, *source diversity*, *class cardinality*, *text type* and *number of tokens*. All constituents have been weighted equally for the calculation of the final confidence value. The influence of the individual constituents is examined in this section. The confidence value has been calculated by omitting one constituent and utilizing only the remaining four constituents. Therefore, five additional runs of the semantic analysis process on the video metadata test set have been performed. These results compared to the results utilizing all constituents are shown in Table 24. The first three columns show recall, precision and F_1-measures when all constituents are applied to calculate the confidence value. The next columns show the results when either *class cardinality*, *source reliability*, *text type*, *source diversity* or *number of tokens* has been omitted for the calculation. Again, the results are divided according to the different sources, entire segments and entire videos.

The evaluation shows that omitting a specific constituent is reflected by the results with respect to recall and/or precision. Also, sometimes no effect is revealed. For example, ignoring the *source diversity* has no effect for the disambiguation of tags. However, the interpretation of these results and subsequent conclusions are not simple. The evaluation shows different results for the specific sources, but

the consequence is not related to the items of the respective source directly. Omitting a specific constituent results in a different order of *all* context items within a context. Therefore, this constituent must be omitted for all context items in order to achieve the presented results. However, the evaluation then shows conflicting results. Whereas omitting the *source diversity* shows no effect for tags, the quality of the results decreases for items from authoritative and ASR sources. Therefore, the influence of the separate constituents requires a detailed observation to draw respective conclusions. This issue will be addressed further in future work.

10.7.7 *Evaluation of the Influence of Negative Context*

The evaluation results of the *conTagger* compared to the *conTagger* using negative context are shown in Table 25. The table presents the results which compare the conTagger using negative context against the same approach without negative context.

The left three columns show recall, precision and F_1-measures for the *conTagger* without taking into account negative context. The right three columns show recall, precision and F_1-measures for the extended version of *conTagger* with negative context for the disambiguation. Recall and precision have been calculated for the metadata items of the different sources (ASR, OCR, user tags and authoritative information) separately and are represented by the respective row.

As shown, recall and precision are slightly improved by using the negative context, especially for metadata items with the lowest prospective confidence values such as OCR metadata. Results for tags and items retrieved by ASR remain nearly constant for both approaches with or without negative context. Overall, these results reveals the actual influence of negative context (although only marginal) in the disambiguation process. The evaluation results will be described in detail in the discussion section.

In addition to the higher recall and precision for context item disambiguation, further evaluation findings include: the increased margin of the disambiguation results.

For this evaluation, the entity candidate with the highest score is chosen as the correctly disambiguated entity for the respective natural language term. The distance between the first and the second highest achieved score for a term can be considered as an indicator the reliability of the achieved result. The higher the distance between the first and the second highest score, the more reliable the result. By using negative context this margin has been increased.

Table 24: Evaluation of the influence of the individual confidence constituents (cc = class cardinality, tt = text type, sd = source diversity, sr = source reliability, nt = number of tokens).

	all			w/o cc			w/o sr			w/o tt			w/o sd			w/o nt		
	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1
Authorative	**60.0**	**54.5**	**57.0**	**60.0**	**54.5**	**57.0**	59.0	53.5	56.0	**60.0**	**54.5**	**57.0**	59.0	54.0	56.5	59.5	54.0	56.5
Tags	**71.0**	**69.5**	**70.0**	70.0	68.5	69.0	**71.0**	69.0	70.0	70.0	69.0	69.5	**71.0**	**69.5**	**70.0**	70.5	68.5	69.5
ASR	55.0	61.0	58.0	**55.5**	61.0	58.0	55.0	61.0	58.0	**55.5**	**61.5**	**58.5**	54.0	60.0	57.0	54.0	60.0	57.0
OCR	56.0	24.0	33.5	57.5	24.0	34.0	**58.0**	24.0	34.0	54.0	21.5	31.0	56.5	24.5	34.0	57.0	**25.0**	**34.5**
Segments	**54.0**	**58.0**	**56.0**	**54.0**	**58.0**	**56.0**	53.5	**58.0**	55.5	**54.0**	**58.0**	**56.0**	53.5	57.5	55.5	53.0	57.5	55.0
Video	**56.0**	**48.0**	**52.0**	**56.0**	**48.0**	**52.0**	55.5	**48.0**	**52.0**	**56.0**	**48.0**	**52.0**	55.5	47.5	51.5	55.0	47.5	51.0

The average margin has been calculated as follows:

$$margin_{average} = \sum_{k=1}^{n} \frac{s_{total(ci_{k,1})} - s_{total(ci_{k,2})}}{n} \quad (21)$$

$s_{total(ci_{k,1})}$ denotes highest achieved total disambiguation score of context item ci_k and $s_{total(ci_{k,2})}$ denotes the second highest achieved disambiguation score for context item ci_k for all context items ci, where $k = 1...n$ and $n = |CI|$, in the reference dataset.

The best results with respect to the margin have been achieved by reducing the *total positive disambiguation score* by the computed *total negative score* (using the weight 1.0 equally for both scores). In case the achieved total disambiguation score is below zero, the total score is set to $s_{total} = 0.0$. Thereby, the interval for the total disambiguation score remains within the range [0.0...1.0].

Table 25: Evaluation results of the *conTagger* compared to *conTagger* using negative context (R = Recall, P = Precision, F_1 = F_1-Measure).

	conTagger			*conTagger Using Negative Context*		
	R	P	F_1	R	P	F_1
Authorative	60.0	54.5	57.0	**61.0**	**55.0**	**58.0**
Tags	71.0	69.5	70.0	71.0	69.5	70.0
ASR	55.0	61.0	58.0	55.0	61.0	58.0
OCR	56.0	24.0	34.0	**60.0**	**26.0**	**36.5**

The averaged margin over all results for the video metadata test set without negative context is 0.27. Taking into account the negative context the margin amounts to 0.40. Therefore, the margin of the analysis results rises at least[9] by an average of 0.13. The increase of the margin is depicted in Table 26 for different sources.

Although, the approach of applying a negative context only shows slight increases of the quality of the results, the influence of negative context has been pointed out. The margin of analysis results is important when the decision for the chosen entity candidates is determined by applying a threshold on the obtained scores. In this case, a higher margin also results in a higher precision.

9 As stated above, negative total disambiguation scores have been avoided; allowing negative total scores might increase the average margin.

Table 26: Increase of margin of analysis results on video metadata test set for different sources and Spotlight dataset.

	Source Type	Margin Without Negative Context	Margin Using Negative Context	Number Of Context Items
Video Dataset	Authoritative	0.43	**0.60**	317
	ASR	0.18	**0.25**	2559
	OCR	0.12	**0.13**	509
	Tags	0.35	**0.60**	536
	All Sources	∅ 0.27	∅ **0.40**	3975
Spotlight Dataset		0.35	**0.45**	409

10.7.8 *Evaluation Using Independent Benchmark*

The context model has been originally developed to process video metadata. Most of the contextual characteristics can only be applied for textual information originating from various sources and of different text types. However, the context model can also be applied for simple natural language texts to achieve a more reliable disambiguation by ordering the text terms. This likewise enables a disambiguation process beginning with the prospectively most confident term. This in turn enables the successive construction of the negative context.

The evaluation results are shown in Table 27. Without using a negative context, the disambiguation approach achieves a recall of 58% and a precision of 41%, resulting in an F_1-measure of 48%. The approach with negative context achieves a recall of 60% and a precision of 42%, resulting in an F_1-measure of 49.5%. In comparison, without any configuration DBpedia Spotlight achieved an F_1-measure of 45%. Additionally, the margin of the disambiguation scores of the highest and second highest score is increased by 0.1 (from 0.35 to 0.45) by using negative context. The positive impact of a negative context is proven on this dataset, although the approach has not been developed specifically for this type of data.

Table 27: Evaluation Results on Spotlight Dataset (R = Recall, P = Precision, F_1 = F_1-measure).

	R	P	F_1
Simple approach without context model	53.0	40.0	45.5
conTagger	58.0	41.0	48.0
conTagger utilizing negative context	**60.0**	**42.0**	**49.5**

10.8 SUMMARY OF EVALUATION RESULTS

This section briefly summarizes the evaluation results presented in the previous sections.

BENCHMARKS Two benchmarks have been created and applied to evaluate the tag processing and the semantic analysis method: one which contains video metadata of four different sources and a second which contains user-generated tags. Additionally, three other benchmarks for evaluation of semantic annotation services have been analyzed.

DICTIONARIES Four different dictionaries for entity look-up have been evaluated. The *RDM* dictionary has been identified as the best trade-off between time complexity for the overall algorithm, mapping coverage and possible recall. This dictionary in its case-insensitive form has been used for subsequent evaluations of the semantic analysis process.

TAG PROCESSING The presented tag processing method achieves up to 39% increased recall and up to 30% increased precision compared to a state-of-the-art annotation service.

DETECTION OF NAMED ENTITIES AND IMPORTANT TERMS The algorithm which detects named entities and important terms in a text has been evaluated on the Spotlight benchmark and the video benchmark. The achieved recall amounts to 72.0% and 80.5%, respectively. Due to the algorithm's *over-annotation* compared to a human annotator, the precision is comparably low with respective values of 48.5% and 55.0%. However, the accuracy of the algorithm is as high as 80%.

DISAMBIGUATION ALGORITHM The disambiguation approach has been evaluated separately from the context model on the Spotlight benchmark against DBpedia Spotlight. The algorithm has been shown to be competitive with the state-of-the-art annotation service.

CONTEXT MODEL The semantic analysis process which applies the context model has been evaluated against the simple segment-based analysis approach on the video benchmark. A significance analysis supports the positive influence of the context model on the analysis results and shows that the results have not been achieved by chance. Additionally, the presented approach has been evaluated against four state-of-the-art annotation approaches. The *conTagger* outperforms the other state-of-the-art tools on almost all measures and for all sources, especially on metadata of sources with low confidence values.

INFLUENCE OF NEGATIVE CONTEXT The novel application of negative context shows only a marginal increase in recall and precision for the analysis results. However, the margin of the disambiguation scores of the entity candidates is increased. This means that the reliability of the interpretation is increased by the application of negative context.

10.9 DISCUSSION

The proposed algorithms have been developed for the specific purpose of semantic analysis and annotation of video metadata. The previous sections presented extensive evaluations of single methods of the overall process as well as of the complete algorithm. Thus, applied dictionaries and benchmarks have been analyzed. The tag processing method as well as the single disambiguation process and conclusively the overall algorithm applying the developed context model and negative context have been evaluated. The achieved results are discussed in the following paragraphs.

The disambiguation approach using the context model achieves improved results regarding recall and precision for context items of all sources. For the evaluation of semantic analysis approaches the focus on either recall or precision is important. In terms of semantic or explorative search based on semantically annotated documents [120], a high precision might be desirable. Incorrectly annotated documents might confuse users of semantic search engines. A high recall, on the other hand, enables a more elaborate explorative search on the annotated documents, because with more annotated entities more relationships can be drawn between documents. The presented evaluation results demonstrate a trade-off between high recall and high precision. By making use of the described confidence value and the disambiguation score, a higher precision – accepting a lower recall – is achievable.

The goal of applying the negative context and negative categories is the elimination of irrelevant topics for the context. Unfortunately, categories do not provide information leading to contextually relevant topics. The categories derived from the Wikipedia (as YAGO, or the Wikipedia classification) do not supply type comprehensive topic information (such as persons, places or organizations belonging to a specific subject). The categories are mostly constricted to one specific type; for example, the category *English Rock Music Groups* is restricted to bands, but does not imply the topic *music*. Thus, the negative context – consisting of negative entities and negative categories – cannot represent negative topics. However, in some cases the disambiguation process has been improved by applying the negative context and devaluing prospectively wrong entity candidates. This shows that negative context can be constructed which partly represents negative top-

ics. Supposedly incorrect entity candidates that also achieved a score within the positive context have been devalued by using the negative context. This is proven by the increase of the recall and the precision.

However, a detailed investigation has shown the impact of the negative context on terms that are not integrated in a context in general. Common or general terms such as *history*, *worry* or *audience* are difficult to disambiguate. As is the case for DBpedia, these terms are often used for band names or music albums and thus, these entities which belong to a very specific category also end up as candidates for common or general terms. The evaluation has shown that these types of entities are often linked to the negative context if the previously disambiguated terms contain such entities as candidates. Such entity candidates are then devalued and can be disregarded in the decision for the correct disambiguation.

11

APPLICATIONS

The context model and the semantic analysis process have been applied as part of a video analysis framework developed and adapted for several projects. The common aim of all projects is the visual and semantic analysis of videos in order to provide a web application which is searchable via entity-centric semantic search. An entity-centric search requires a specific search interface to provide the user with relevant entities for the entered search string. Thus, the user is asked to determine the specific meaning of the entered search string by choosing an entity from a suggested list. This process is called *autosuggestion*. Similar to Google's instant search, the user interface provides a list of entities relevant for the current search string. The list is faceted into persons, organizations, events and other things according to the types of the suggested entities.

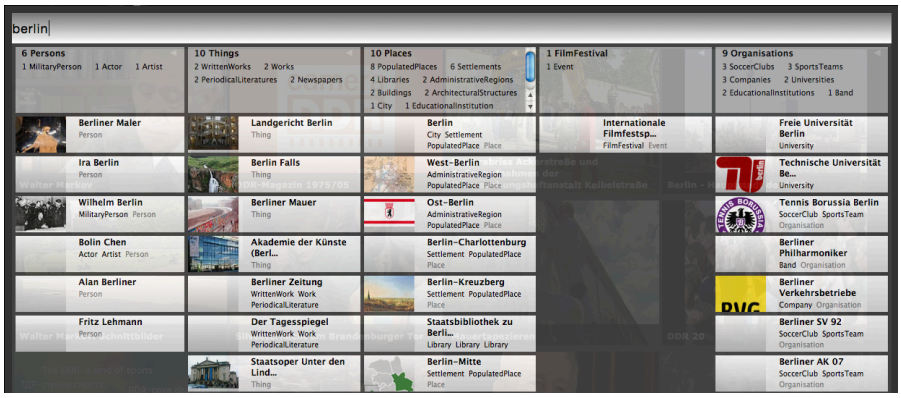


Figure 25: Screenshot of faceted list of suggested entities for the search string *berlin* within the *mediaglobe* user interface.

Figure 25 shows a faceted suggestion for the search string *berlin*. On the basis of this list, the user can decide which entity should be the target of the search. The subsequent list of results contains only documents that have been annotated with the selected entity by the semantic analysis of the metadata. The autosuggestion is based on a dictionary containing labels and popularity scores of the respective entities. The underlying suggestion index is thus organized as a list of available entities. For every entity, a list of labels and their respective relevance for the entity as well as the ontology type, a thumbnail (if one exists), and the main label (optionally multilingual) are provided. For the projects introduced in the following sections, the suggestion index has been created based on the *RDM* dictionary introduced in Section 10.5.4.

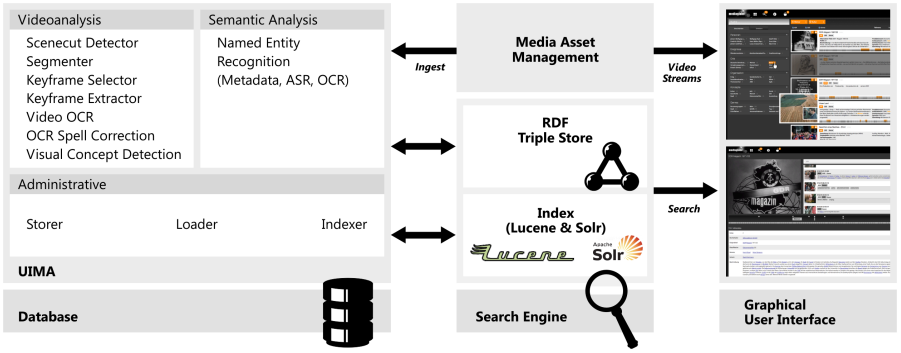


Figure 26: Overall architecture of *mediaglobe* [48]

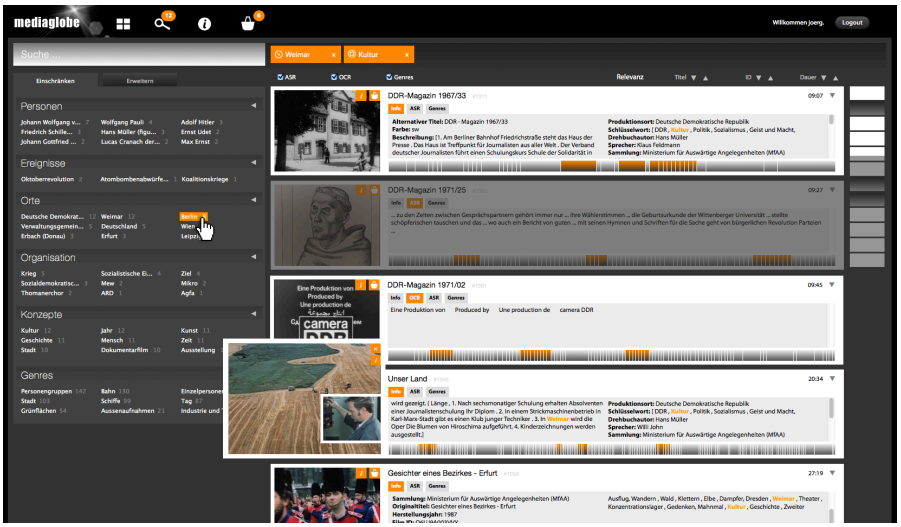


Figure 27: Screenshot of *mediaglobe* user interface.

11.1 MEDIAGLOBE

The *mediaglobe* research project[1] was initialized in late 2009 by the project partners *transfer Media*[2], *Defa Spektrum*[3], *Flow Works*[4] and the *Hasso Plattner Institute* and was concluded in May 2012. It was part of the THESEUS research program[5] funded by the German Federal Ministry for Economics and Technology. The primary goal of *mediaglobe* was to develop a generally applicable infrastructure for the retrieval of audiovisual archives with an emphasis on historical documentaries, for the purpose of making cultural heritage documents widely accessible [48]. The underlying archive comprises documentary films of the former German Democratic Republic.

As shown in Figure 26, the analysis process includes several visual and semantic analyses. Textual information is extracted and provided

1 http://www.projekt-mediaglobe.de.
2 http://www.transfermedia.de/
3 http://www.defa-spektrum.de/
4 http://www.flowworks.de
5 http://theseus.pt-dlr.de/en/

for a semantic analysis. The results are stored in a triple store and indexed for a semantic search, and finally provided in a user interface. A screen shot of the user interface is shown in Figure 27. For the semantic analysis process, an early version of the context model and the disambiguation process has been applied. DBpedia has been utilized for the semantic annotation.

11.2 SEMEX

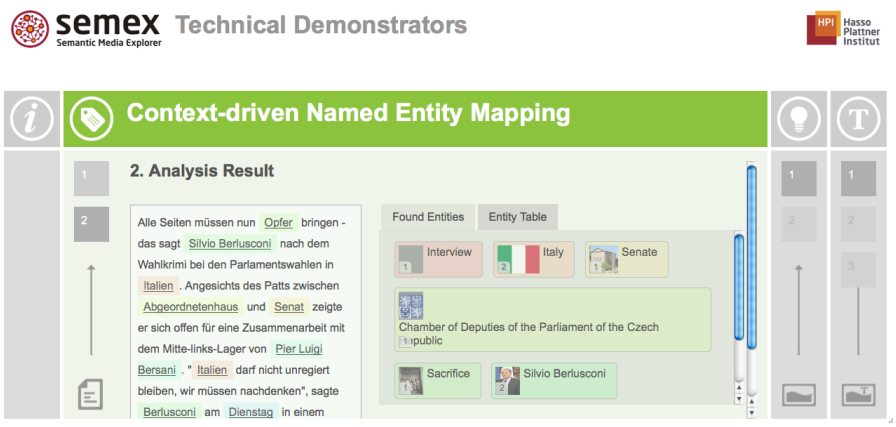


Figure 28: Screenshot of technical demonstrator of the semantic analysis included in the SeMEx.

SeMEx stands for *Semantic Media Explorer* and is based on the analysis prototype developed for the *mediaglobe* project. For this internal project at the Hasso Plattner Institute, the visual as well as semantic analysis methods are further developed and enhanced [93]. The context model of *conTagger* has been developed and evaluated in the scope of this project.

The scientific results and innovative achievements of SeMEx have been presented at conferences, fairs and to professionals from industry and research. As a result, a web application has been developed introducing the basics of the analysis processes separately. A screenshot of the demonstrator for the semantic analysis is shown in Figure 28. The main focus of the semantic analysis process within the SeMEx project lies in the analysis of video metadata using the DBpedia as its underlying knowledge base.

11.3 AV PORTAL

The AV Portal is a joint project of the *German National Library of Science and Technology*[6], Flow Works and the Hasso Plattner Institute. The project strives to enhance the access to non-textual media in the scope

6 http://www.tib.uni-hannover.de

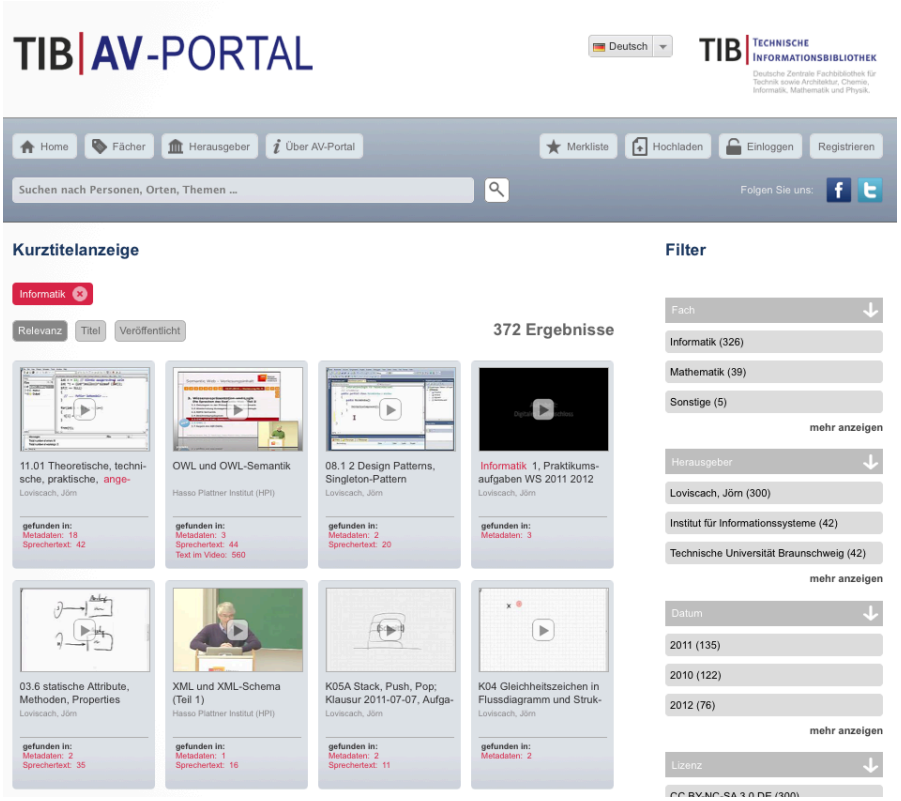


Figure 29: Screenshot of the user interface of the AV Portal.

of library services. In particular, the AV Portal provides an entity-based semantic search of scientific videos from the fields of technology and natural sciences. Figure 29 shows a screenshot of the user interface.

The AV Portal utilizes the GND as its underlying knowledge base. The provided videos are primarily assigned to a specific subject area. Following project requirements, the videos are only annotated with concepts belonging to the subject area to which the video is assigned. Therefore, several domain experts defined excerpts of the entire knowledge base related with the subject areas. For the semantic analysis, all named entities detected in the textual information and the entire knowledge base are taken into account for the disambiguation process. Only concepts which belong to the respective subject area are maintained and provided for the semantic search.

As shown in the previous sections, the presented semantic analysis method has been applied successfully for several projects. All projects strive to provide semantically annotated videos for a semantic search. Nevertheless, the presented approach can be applied for any type of video archive, using any underlying knowledge base. Also, the context model can be applied for simple continuous text as well as for any documentary metadata that originates from different sources.

12 CONCLUSION AND OUTLOOK

This chapter concludes the thesis with a summary of the achieved results and scientific contributions as well as an outlook toward possible future work.

12.1 SUMMARY

Semantic annotation provides essential insight into the content of documents under consideration during semantic analysis. In many cases, the sheer amount of documents which must be processed does not allow for manual annotation. State-of-the-art semantic annotation tools focus on the annotation of simple web documents containing continuous text. The textual information in such documents is considered to be homogenous: all information originates from the same source and possesses the same characteristics. Therefore, all information contained in such a document might be processed equally and the context is also considered homogenous. However, this is not the case for video documents or any other document annotated with metadata of different types and origins. Semantic annotation of videos requires special attention. Initially, textual content-based information must be extracted by automatic algorithms such as OCR and ASR. Additionally, authoritative metadata, such as titles and descriptive texts, might be provided. Therefore, content-based metadata of videos consists of different metadata types possessing different characteristics and reliabilities. The context for video documents which contain this metadata is thereby considered to be heterogenous.

This thesis has addressed the problem of semantic analysis of video metadata taking into account heterogenous contexts. For that purpose, a context model has been developed which considers the characteristics of different metadata information and subsequently derives a confidence value. The confidence value reflects the level of correctness and prospective ambiguity of the information. The confidence is highest the greater the assumed correctness and the lower the ambiguity. The metadata items are ordered according to their confidence value and the analysis process is performed in descending order starting with the item of the highest confidence value. Under the assumption that items with a high confidence value will be interpreted correctly, previously processed items serve as reference points for subsequent disambiguation processes.

The thesis also presents a disambiguation algorithm which considers descriptive texts and semantic relationships to distinguish differ-

ent entities. Additionally, a coreference analysis and several heuristics are presented that can be used for the disambiguation of ambiguous textual information. A disambiguation can only be performed successfully when a context is provided that supports the interpretation. For the presented approach, the context for the disambiguation is created dynamically for each item considering their confidence values and other characteristics. A novel concept presented in this thesis is the negative context. This type of context represents information that is not relevant for an interpretation. Within the Semantic Web, the open world assumption is applied. Therefore, missing information cannot be treated as wrong or irrelevant. The negative context is created during the disambiguation process and includes all information that has been identified as irrelevant or contradictory to previous disambiguations.

The developed algorithms have been evaluated on several benchmarks and against several state-of-the-art tools. The presented algorithms show improved results with respect to the quality of the annotations; the results for metadata originating from sources of lower reliability have been especially improved using the context model. Prior to the actual evaluations, several benchmarks and dictionaries for entity look-up were analyzed and evaluated. Additionally, a new benchmark for the special purpose of semantic analysis of video metadata has been created and introduced.

The presented approach is integrated into a video analysis framework that has been successfully applied in several projects for the purpose of a content-based semantic search. For the practical application of the semantic analysis process, two different knowledge bases have been used. This shows the practicability and flexibility of the presented approach.

12.2 FUTURE WORK

The semantic analysis process presented in this thesis has been fully implemented and integrated into a video analysis framework. Future work will focus on the enhancement of single methods and thereby the improvement of analysis results.

Ongoing work includes research on building the negative context by using latent topics. Boehm et al. describe an approach on aggregating the entities of an RDF graph into subgraphs and thereby constructing latent topics [13]. The entities of such subgraphs can be appended to the negative context if a sufficient number of previously eliminated *negative entities* is included in such a graph. This approach removes specific topics from the list of relevant topics for the present positive context.

A context can also further be refined by sampling low-level adjustments, such as white lists. Whitelists can be created either statically

by applying a specific knowledge base that only contains relevant entities and reduces the ambiguity of terms, or by logical constraints in terms of rules. For example, a document published in 1960 most likely references only persons born before this date. Thereby, only a restricted number of available semantic entities qualify for the analysis of such a document. While persons naturally have a time reference, other real world entities may be hard to classify (see Section 6.3.2 for aggregation of entities according to a time reference). Ongoing work might include the definition of a time-related scope for various entity types and further types of restrictions. Along the lines of the theory that semantic entities are time-referenced, the research field of Named Entity Evolution (NEE) has emerged [72, 111]. The characteristics of semantic entities might change over time. New countries are founded, others disappear or are re-named, and borders are shifted. A person with a political career becomes senator, then a president, and finally a civilian again. All of these changes must be taken into account for a semantic analysis of time-referenced textual information, when considering the appropriate context. Thus, NEE will also be part of future research on a context-based semantic analysis.

The Link Graph Analysis (see Section 8.2.4) can be a time-consuming approach if the underlying graph is very large. For this thesis, the approach has been implemented in various ways to identify the fastest algorithm for the detection of paths in a large graph. Adjacent matrices, pre-processed closure and index storage, dynamic creation of sub-graphs according to the growing context, etc. Future work might evaluate several approaches to identify the fastest method for context-aware semantic analysis.

The evaluation of the importance of the separate analysis methods included in the disambiguation process has been performed for a specific knowledge base and benchmark (see Section 10.7.4). Future work might include a determination of the score weights by the application of machine learning algorithms. The weights might be identified for knowledge bases that provide less or different information compared to the DBpedia and for other benchmarks.

These enhancements planned as future work aim to improve the achieved analysis results and to further develop context characteristics. However, this thesis presents an approach that pays particular attention to the characteristics of metadata that originate from different sources. The developed context model and disambiguation approach have been evaluated extensively on different benchmarks. The proposed hypotheses have been confirmed and supported by the high quality of the results. Although this thesis focuses on video metadata, the approach can be adapted to other document types that feature metadata from multiple sources and various reliabilities.

Part IV

APPENDIX

A

APPENDIX

The following tables present additional statistics about the dictionaries presented in Section 10.5 based on the five different benchmarks presented in Section 10.3.

Table 28: Case-Sensitive: Recall and Precision, if most popular entity – based on *incoming Wikipedia page links* – is mapped to mention

BM \ Dic			SPL		RDM		AIDA		GCW		Mention Count
			2M entries		10M entries		18M entries		378M entries		
Spotlight		R	149	56%	50	19%	36	14%	27	10%	265
		P		63%		77%		16%		10%	
KORE 50		R	49	38%	50	38%	56	43%	20	15%	130
		P		42%		39%		44%		15%	
Wikilinks		R	77,583	40%	81,259	42%	75,104	39%	90,458	47%	192,008
		P		72%		71%		65%		53%	
Video Dataset	Authoritative	R	172	52%	64	19%	52	16%	40	12%	330
		P		61%		63%		19%		13%	
	Tags	R	140	43%	3	1%	12	4%	76	23%	329
		P		59%		100%		5%		24%	
	OCR	R	64	47%	47	34%	25	18%	20	15%	137
		P		65%		66%		21%		16%	
	ASR	R	948	52%	180	10%	216	12%	185	10%	1,812
		P		59%		72%		14%		10%	
Tag Dataset		R	158	31%	10	2%	28	6%	139	28%	505
		P		59%		91%		11%		30%	

Table 29: Case-Insensitive: Recall and Precision, if most popular entity – based on *incoming Wikipedia page links* – is mapped to mention

BM \ Dic			SPL		RDM		AIDA		GCW		Mention Count
			2M entries		10M entries		18M entries		378M entries		
Spotlight		R	129	49%	154	58%	43	16%	26	10%	265
		P		54%		62%		18%		10%	
KORE 50		R	50	38%	50	38%	56	43%	18	14%	130
		P		41%		38%		43%		14%	
Wikilinks		R	81,424	42%	100,179	52%	83,949	44%	85,805	45%	192,008
		P		71%		69%		66%		50%	
Video Dataset	Authoritative	R	148	45%	176	53%	68	21%	39	12%	330
		P		52%		57%		22%		12%	
	Tags	R	154	47%	194	59%	68	21%	46	14%	329
		P		52%		62%		22%		14%	
	OCR	R	71	52%	82	60%	29	21%	16	12%	137
		P		61%		62%		23%		12%	
	ASR	R	847	47%	1,012	56%	243	13%	158	9%	1,812
		P		51%		58%		14%		9%	
Tag Dataset		R	197	39%	230	46%	107	21%	89	18%	505
		P		53%		54%		27%		18%	

Table 30: Recall and Precision, if most popular entity – based on *Google popularity* for mention as anchor for entity – is mapped to mention

BM \ Dic			SPL		RDM		AIDA		GCW		Mention Count
			2M entries		10M entries		18M entries		378M entries		
Spotlight		R	199	75%	55	21%	51	19%	187	71%	265
		P		85%		85%		22%		72%	
KORE 50		R	50	38%	56	43%	59	45%	40	31%	130
		P		43%		43%		46%		31%	
Wikilinks		R	79,235	41%	83,079	43%	78,638	41%	120,225	63%	192,008
		P		74%		73%		68%		70%	
Video Dataset	Authoritative	R	219	66%	75	23%	70	21%	223	68%	330
		P		78%		74%		25%		71%	
	Tags	R	180	55%	3	1%	18	5%	224	68%	329
		P		76%		100%		8%		71%	
	OCR	R	74	54%	59	43%	30	22%	90	66%	137
		P		76%		83%		25%		71%	
	ASR	R	1,227	68%	197	11%	247	14%	1187	66%	1,812
		P		14%		78%		15%		67%	
Tag Dataset		R	191	38%	10	2%	32	6%	292	58%	505
		P		71%		91%		12%		63%	

Table 31: Recall and precision, if most popular entity – based on *Spotlight popularity* for mention as anchor for entity – is mapped to mention

BM \ Dic			SPL 2M entries		RDM 10M entries		AIDA 18M entries		GCW 378M entries		Mention Count
Spotlight		R	200	75%	53	20%	51	19%	205	77%	265
		P		85%		82%		22%		79%	
KORE 50		R	36	28%	37	28%	43	33%	43	33%	130
		P		31%		29%		34%		33%	
Wikilinks		R	79,226	41%	82,469	43%	78,073	41%	119,925	62%	192,008
		P		74%		72%		68%		70%	
Video Dataset	Authoritative	R	219	66%	72	22%	70	21%	175	53%	330
		P		78%		71%		25%		56%	
	Tags	R	177	54%	3	1%	18	5%	182	55%	329
		P		74%		100%		8%		58%	
	OCR	R	74	54%	56	41%	32	23%	78	57%	137
		P		76%		79%		27%		61%	
	ASR	R	1,234	68%	197	11%	259	14%	949	52%	1,812
		P		77%		78%		16%		53%	
Tag Dataset		R	193	38%	10	2%	32	6%	313	62%	505
		P		72%		91%		12%		67%	

BIBLIOGRAPHY

[1] Gregory D. Abowd and Elizabeth D. Mynatt. Charting past, present, and future research in ubiquitous computing. ACM Transactions on Computer-Human Interaction, 7(1):29–58, March 2000. ISSN 1073-0516. doi: 10.1145/344949.344988. URL http://doi.acm.org/10.1145/344949.344988.

[2] D. Adjeroh, M. C. Lee, N. Banda, and U. Kandaswamy. Adaptive edge-oriented shot boundary detection. Journal on Image and Video Processing, pages 5:1–5:13, January 2009. ISSN 1687-5176. doi: 10.1155/2009/859371. URL http://dx.doi.org/10.1155/2009/859371.

[3] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In Proceedings of the 2008 ACM conference on Recommender systems, RecSys '08, pages 335–336, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-093-7. doi: 10.1145/1454008.1454068. URL http://doi.acm.org/10.1145/1454008.1454068.

[4] Adrian Akmajian, Richard A. Demers, Ann K. Farmer, and Robert M. Harnish. Linguistics: An introduction to Language and Communication. MIT Press, Cambridge, Massachusetts, 2001.

[5] A. M. Amel, A. B. Abdelali, and A. Mtibaa. Video shot boundary detection using motion activity descriptor. Journal of Documentation of Telecommunications, 2(1):54–59, 2010.

[6] Peter Auer. Kontextualisierung. In Studium Linguistik. Albert-Ludwigs-Universität Freiburg, 1986.

[7] Leonard E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In Oved Shisha, editor, Inequalities III: Proceedings of the Third Symposium on Inequalities, pages 1–8, University of California, Los Angeles, 1972. Academic Press.

[8] Mary Bazire and Patrick Brézillon. Understanding context before using it. In Proceedings of the 5th international conference on Modeling and Using Context, CONTEXT'05, pages 29–40, Berlin, Heidelberg, 2005. Springer. ISBN 3-540-26924-X, 978-3-540-26924-3. doi: 10.1007/11508373_3. URL http://dx.doi.org/10.1007/11508373_3.

[9] Mike Bergman. Announcing umbel: A lightweight subject structure for the web, 2007. URL http://tinyurl.com/o5k5yq3. last visited on November, 17th 2013.

[10] Tim Berners-Lee. Notation 3 logic – an rdf language for the semantic web, 1998. URL http://www.w3.org/DesignIssues/Notation3.html. Last visited on November, 13th 2013.

[11] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. Scientific American, 284(5):34–43, 2001. URL http://www.sciam.com/article.cfm?id=the-semantic-web.

[12] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. Scientific American, pages 29–37, 2001.

[13] Christoph Böhm, Gjergji Kasneci, and Felix Naumann. Latent topics in graph-structured data. In Proceedings of the 21st ACM international conference on Information and knowledge management, CIKM '12, pages 2663–2666, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2398718. URL http://doi.acm.org/10.1145/2396761.2398718.

[14] Stefan Bordag. Word sense induction: Triplet-based clustering and automatic evaluation. In In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Trento, Italy, 2006. Association for Computational Linguistics.

[15] Andrew Eliot Borthwick. A Maximum Entropy Approach to Named Entity Recognition. PhD thesis, New York University, 1999.

[16] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Computational Linguistics, 21(4):543–566, 1995.

[17] Eric Brill. Part-of-speech tagging. In Handbook of Natural Language Processing. Marcel Dekker, Inc., 2000.

[18] Volha Bryl, Claudio Giuliano, Luciano Serafini, and Kateryna Tymoshenko. Supporting natural language processing with background knowledge: coreference resolution case. In Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I, ISWC'10, pages 80–95, Berlin, Heidelberg, 2010. Springer. ISBN 3-642-17745-X, 978-3-642-17745-3. URL http://dl.acm.org/citation.cfm?id=1940281.1940288.

[19] Kenneth Ward Church. A stochastic parts program and noun phrase parser for unrestricted text. In ANLP, pages 136–143.

Association for Computational Linguistics, 1988. URL http://dblp.uni-trier.de/db/conf/anlp/anlp1988.html#Church88.

[20] Garrison W. Cottrell. A connectionist approach to word sense disambiguation. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989. ISBN 0-934613-61-3.

[21] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray, and D. Maupertuis. Visual categorization with bags of keypoints. In Workshop on Statistical Learning in Computer Vision, pages 1–22. Springer, 2004.

[22] Danica Damljanovic and Kalina Bontcheva. Named entity disambiguation using linked data. In 9th Extended Semantic Web Conference (ESWC2012). Springer, May 2012. URL http://data.semanticweb.org/conference/eswc/2012/paper/poster/334.

[23] John Dewey. Context and thought. University of California publications in philosophy, 12, 1931.

[24] Mostefa Djamel, Hamon Olivier, and Choukri Khalid. Evaluation of automatic speech recognition and speech language translation within tc-star : Results from the first evaluation campaign. In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC06), Genoa, Italy, May 2006. European Language Resources Association (ELRA).

[25] Judith Domínguez-Borràs, Manuel Garcia-Garcia, and Carles Escera. Negative emotional context enhances auditory novelty processing: behavioural and electrophysiological evidence. European Journal of Neuroscience, 28(6):1199–1206, 2008. ISSN 1460-9568. doi: 10.1111/j.1460-9568.2008.06411.x. URL http://dx.doi.org/10.1111/j.1460-9568.2008.06411.x.

[26] Beate Dorow and Dominic Widdows. Discovering corpus-specific word senses. In Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 2, EACL '03, pages 79–82, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. ISBN 1-111-56789-0. doi: 10.3115/1067737.1067753. URL http://dx.doi.org/10.3115/1067737.1067753.

[27] Paul Dourish. What we talk about when we talk about context. Personal Ubiquitous Comput., 8(1):19–30, February 2004. ISSN 1617-4909. doi: 10.1007/s00779-003-0253-8. URL http://dx.doi.org/10.1007/s00779-003-0253-8.

[28] Bradley Efron and Robert J. Tibshirani. An introduction to the bootstrap. Chapman & Hall, 1993.

[29] Christine Englund. Speech recognition in the jas 39 gripen aircraft - adaptation to speech at different g-loads. Master's thesis, Royal Institute of Technology, Stockholm, 2004.

[30] Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An, editors, Proceedings of CIKM 2010, pages 1625–1628. ACM, 2010. ISBN 978-1-4503-0099-5. URL http://dblp.uni-trier.de/db/conf/cikm/cikm2010.html#FerraginaS10.

[31] 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. doi: 10.3115/1219840.1219885. URL http://dx.doi.org/10.3115/1219840.1219885.

[32] Ovidiu Fortu and Dan Moldovan. Identification of textual contexts. In Proceedings of the 5th international conference on Modeling and Using Context, CONTEXT'05, pages 169–182, Berlin, Heidelberg, 2005. Springer. ISBN 3-540-26924-X, 978-3-540-26924-3. doi: 10.1007/11508373_13. URL http://dx.doi.org/10.1007/11508373_13.

[33] Winthrop Nelson Francis and Henry Kucera. Frequency Analysis of English Usage: Lexicon and Grammar. Houghton Mifflin, 1983.

[34] Gottlob Frege. The Foundations of Arithmetic. Northwestern University Press, Evanston, Illinois, 1884/1980.

[35] WilliamA. Gale, KennethW. Church, and David Yarowsky. A method for disambiguating word senses in a large corpus. Computers and the Humanities, 26(5-6):415–439, 1992. ISSN 0010-4817. doi: 10.1007/BF00136984. URL http://dx.doi.org/10.1007/BF00136984.

[36] Aldo Gangemi. A comparison of knowledge extraction tools for the semantic web. In Philipp Cimiano, Óscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, ESWC, volume 7882 of Lecture Notes in Computer Science, pages 351–366. Springer, 2013. ISBN 978-3-642-38287-1, 978-3-642-38288-8.

[37] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. Computational Linguistics., 28 (3):245–288, September 2002. ISSN 0891-2017. doi: 10.

1162/089120102760275983. URL http://dx.doi.org/10.1162/089120102760275983.

[38] Dan Gillman. Iso/iec 11179-1 information technology – metadata registries (mdr) - part 1: Framework ed 3, 2012. URL http://metadata-standards.org/11179/#A1. Committee Draft 1.

[39] Dan Gillman. Iso/iec 11179-1 information technology – metadata registries (mdr) - part 1: Framework ed 3, 2013. URL http://metadata-standards.org/11179/#A1. Committee Draft 2.

[40] Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. Journal of Information Science, 32:198–208, April 2006. ISSN 0165-5515. doi: 10.1177/0165551506062337. URL http://portal.acm.org/citation.cfm?id=1119738.1119747.

[41] Gregory Grefenstette and Pasi Tapanainen. What is a word, what is a sentence? problems of tokenization. In Proceedings of 3rd International Conference on Computational Lexicography (COMPLEX), pages 79–87, 1994.

[42] Thomas Grill and Manfred Tscheligi. Towards a multi-perspectival approach of describing context. In Michael Beigl, Henning Christiansen, ThomasR. Roth-Berghofer, Anders Kofod-Petersen, KennyR. Coventry, and HeddaR. Schmidtke, editors, Modeling and Using Context, volume 6967 of Lecture Notes in Computer Science, pages 115–118. Springer, 2011. ISBN 978-3-642-24278-6. doi: 10.1007/978-3-642-24279-3_13. URL http://dx.doi.org/10.1007/978-3-642-24279-3_13.

[43] Steven Gross. Essays on Linguistic Context-Sensitivity and its Philosophical Significance. Routledge & Kegan Paul, 2001.

[44] T.R. Gruber. A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2):199–220, 1993.

[45] John Gumperz and Jenny Cook-Gumperz. Context in children's speech. In Papers on Language and Context. University of California press, 1976.

[46] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. Evaluating entity linking with wikipedia. Artificial Intelligence, 194:130–150, 2012. URL http://dblp.uni-trier.de/db/journals/ai/ai194.html#HacheyRNHC13.

[47] Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating nlp using linked data. In Proceedings of

12th International Semantic Web Conference, Sydney, Australia, October 2013. Springer.

[48] C. Hentschel, J. Hercher, M. Knuth, J. Osterhoff, B. Quehl, H. Sack, N. Steinmetz, J. Waitelonis, and H.Yang. Open up cultural heritage in video archives with mediaglobe. In Proc. of 12th International Conference on Innovative Internet Community Services (I2CS 2012), number 204 in Lecture Notes in Informatics (LNI). Springer, 2012.

[49] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. Foundations of Semantic Web Technologies. Chapman & Hall, 2009.

[50] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In Proc. of the Conf. on Empirical Methods in Natural Language Processing, EMNLP '11, pages 782–792, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL http://dl.acm.org/citation.cfm?id=2145432.2145521.

[51] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. KORE: Keyphrase overlap relatedness for entity disambiguation. In Proceedings of the 21st ACM international conference on Information and knowledge management, pages 545–554. ACM, ACM, 2012.

[52] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 28(1), 1972. URL http://www.soi.city.ac.uk/~ser/idfpapers/ksj_orig.pdf.

[53] Daniel Jurafsky and James H. Martin. Speech and Language Processing. Prentice Hall, 2000.

[54] Lauri Karttunen. Discourse referents. In Syntax and Semantics 7: Notes from the Linguistic Underground. Academic Press, 1976.

[55] S. Klein and R. Simmons. A computational approach to grammatical coding of english words. ACM Computer Survey, 10: 334–347, 1963.

[56] 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. ISBN 1-55860-778-1. URL http://dl.acm.org/citation.cfm?id=645530.655813.

[57] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computational Linguistics, 2013.

[58] Yoong Keok Lee and Hwee Tou Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10, EMNLP '02, pages 41–48, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118699. URL http://dx.doi.org/10.3115/1118693.1118699.

[59] Timothy Robert Leek. Information extraction using hidden markov models. Master's thesis, University of California, San Diego, 1997.

[60] Douglas Lenat. The dimensions of context-space. Cycorp technical report, Cycorp, 1998.

[61] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the Fifth International Conference on Systems Documentation. ACM, 1986.

[62] David Lewis. General semantics. Synthese, 22(1-2):18–67, December 1970. ISSN 0039-7857. doi: 10.1007/BF00413598. URL http://dx.doi.org/10.1007/BF00413598.

[63] Nadine Ludwig and Harald Sack. Named entity recognition for user-generated tags. In Proceedings of the 2011 22nd International Workshop on Database and Expert Systems Applications, DEXA '11, pages 177–181, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4486-1. doi: 10.1109/DEXA.2011.56. URL http://dx.doi.org/10.1109/DEXA.2011.56.

[64] Nadine Ludwig, Jörg Waitelonis, Magnus Knuth, and Harald Sack. WhoKnows? - evaluating linked data heuristics with a quiz that cleans up dbpedia. In Proceedings of the 8th Extended Semantic Web Conference (ESWC), Poster-Session. Springer, 2011.

[65] John C. Mallery. Thinking about foreign policy: Finding an appropriate role for artificially intelligent computers. Master's thesis, Massachussetts Institute of Technology, Cambridge, MA, 1988.

[66] Christopher D. Manning. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part I, CICLing'11, pages 171–189, Berlin, Heidelberg, 2011. Springer. ISBN 978-3-642-19399-6. URL http://dl.acm.org/citation.cfm?id=1964799.1964816.

[67] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. Computational Linguistics, 19(2):313–330, 1993.

[68] Jean-Pierre Martens. Continuous speech recognition over the telephone. Technical report, University of Gent, 2000.

[69] Mark T. Maybury. Introduction. In Multimedia Information Éxtraction - Advances in Video, Audio, and Imafery Analysis for Search, Data Mining, Surveillance, and Authoring. Wiley / IEEE Computer Society, 2013.

[70] Diana Maynard. In defence of sentiment analysis: The wrong kind of snow. http://tinyurl.com/njbotbv, October 2013. Last visited on November, 17th 2013.

[71] Diana Maynard, Kalina Bontcheva, and Dominic Rout. Challenges in developing opinion mining tools for social media. In Proceedings of @NLP Workshop at LREC 2012, Istanbul, Turkey, 2012. European Language Resources Association (ELRA).

[72] Arturas Mazeika, Tomasz Tylenda, and Gerhard Weikum. Entity timelines: visual analytics and named entity evolution. In Proceedings of the 20th ACM international conference on Information and knowledge management, CIKM '11, pages 2585–2588, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0717-8. doi: 10.1145/2063576.2064026. URL http://doi.acm.org/10.1145/2063576.2064026.

[73] Patrick McCrae. A Computational Model for the Influence of Cross-Modal Context upon Syntactic Parsing. PhD thesis, Universität Hamburg, Von-Melle-Park 3, 20146 Hamburg, 2010. URL http://ediss.sub.uni-hamburg.de/volltexte/2010/4800.

[74] Pankaj Mehra. Context-aware computing: Beyond search and location-based services. IEEE Internet Computing, 16:12–16, 2012. ISSN 1089-7801. doi: http://doi.ieeecomputersociety.org/10.1109/MIC.2012.31.

[75] 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. ISBN 978-1-4503-0621-8. doi: 10.1145/2063518.2063519. URL http://doi.acm.org/10.1145/2063518.2063519.

[76] Rada Mihalcea. Co-training and self-training for word sense disambiguation. In Hwee, editor, HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning, pages 33–40. Association for Computational Linguistics, May 2004.

[77] 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. ISBN 978-1-59593-803-9. doi: 10.1145/1321440.1321475. URL http://doi.acm.org/10.1145/1321440.1321475.

[78] 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. ISBN 978-1-59593-991-3. doi: 10.1145/1458082.1458150. URL http://doi.acm.org/10.1145/1458082.1458150.

[79] Thomas M. Mitchell. Machine Learning. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

[80] William Morgan. Statistical hypothesis tests for nlp, 2013. URL http://masanjin.net/sigtest.pdf. Last visited on November, 13th 2013.

[81] Charles W. Morris. Foundations of the theory of signs. University of Chicago press, 1938.

[82] Roberto Navigli. Word sense disambiguation: A survey. ACM Computer Survey, 41(2):10:1–10:69, February 2009. ISSN 0360-0300. doi: 10.1145/1459352.1459355. URL http://doi.acm.org/10.1145/1459352.1459355.

[83] Hwee T. Ng. Getting serious about word sense disambiguation. In Proceedings of the SIGLEX Workshop. Association for Computational Linguistics, 1997.

[84] Eric W. Noreen. Computer Intensive Methods for Testing Hypothesis. John Wiley & Sons, 1989.

[85] Oded Nov and Chen Ye. Why do people tag?: motivations for photo tagging. Communications of the ACM, 53:128–131, July 2010. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/1785414.1785450. URL http://doi.acm.org/10.1145/1785414.1785450.

[86] Library of Congress Digital Repository Development. Core metadata elements. Technical report, Library of Congress, 1998. URL http://www.loc.gov/standards/metadata.html. Last visited on November, 13th 2013.

[87] Charles Kay Ogden and Ivor Armstrong Richards. The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism. Routledge & Kegan Paul, 1923.

[88] David D. Palmer. Tokenisation and sentence segmentation. In Handbook of Natural Language Processing. Marcel Dekker, Inc., 2000.

[89] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. Foundation of Trends in Information Retrieval, 2(1-2):1–135, January 2008. ISSN 1554-0669. doi: 10.1561/1500000011. URL http://dx.doi.org/10.1561/1500000011.

[90] Massimo Poesio. Semantic analysis. In Handbook of Natural Language Processing. Marcel Dekker, Inc., 2000.

[91] J. Ross Quinlan. C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

[92] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics, 19(1):17–30, 1989.

[93] Harald Sack. Semex: Enabling exploratory video search by semantic video analysis. In Working Notes of the LWA 2011 - Learning, Knowledge, Adaptation, 2011.

[94] Neela Sawant, Jia Li, and James Ze Wang. Automatic image semantic interpretation using social action and tagging data. Multimedia Tools Appl., 51(1):213–246, 2011. URL http://dblp.uni-trier.de/db/journals/mta/mta51.html#SawantLW11.

[95] Albrecht Schmidt, Michael Beigl, and Hans-W Gellersen. There is more to context than location. Computers & Graphics, 23 (6):893 – 901, 1999. ISSN 0097-8493. doi: http://dx.doi.org/10.1016/S0097-8493(99)00120-X. URL http://www.sciencedirect.com/science/article/pii/S009784939900120X.

[96] Ingo Schmitt. Ähnlichkeitssuche in Multimedia-Datenbanken: Retrieval, Suchalgorithmen und Anfragebehandlung. Oldenbourg, 2006. ISBN 3-486-57907-X. 445 pages.

[97] Hinrich Schütze. Automatic word sense discrimination. Computational Linguistics, 24(1):97–124, 1998.

[98] Satoshi Sekine. Named entity: History and future. Technical report, New York University, 2004.

[99] Prithviraj Sen. Collective context-aware topic models for entity disambiguation. In Proceedings of the 21st international conference on World Wide Web, WWW '12, pages 729–738, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187935. URL http://doi.acm.org/10.1145/2187836.2187935.

[100] Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015, University of Massachusetts Amherst, 2012.

[101] John F. Sowa. Syntax, semantics, and pragmatics of contexts. Technical report, State University of New York at Binghamton, 1995.

[102] Louise F. Spiteri. The use of collaborative tagging in public library catalogues. Proceedings of the American Society for Information Science and Technology, 43(1):1–5, 2006. ISSN 1550-8390. doi: 10.1002/meet.14504301214. URL http://dx.doi.org/10.1002/meet.14504301214.

[103] Valentin I. Spitkovsky and Angel X. Chang. A cross-lingual dictionary for english wikipedia concepts. In Nicoletta Calzolari (Conf. Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, Proc. of the Eight Int. Conf. on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

[104] Steffen Staab and Rudi Studer, editors. Handbook on Ontologies. International Handbooks on Information Systems. Springer, 2004. ISBN 3-540-40834-7.

[105] Paul Stapleton and Rena Helms-Park. Evaluating web sources in an eap course: Introducing a multi-trait instrument for feedback and assessment. English for Specific Purposes, 25(4): 438 – 455, 2006. ISSN 0889-4906. doi: http://dx.doi.org/10.

1016/j.esp.2005.11.001. URL http://www.sciencedirect.com/science/article/pii/S0889490605000700.

[106] Thomas Steiner. Semwebvid - making video a first class semantic web citizen and a first class web bourgeois. In Axel Polleres and Huajun Chen, editors, ISWC Posters&Demos, volume 658 of CEUR Workshop Proceedings. CEUR-WS.org, 2010. URL http://dblp.uni-trier.de/db/conf/semweb/pd2010.html#Steiner10.

[107] Nadine Steinmetz and Harald Sack. Semantic multimedia information retrieval based on contextual descriptions. In Philipp Cimiano, Óscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, Proceedings of the 10th Extended Semantic Web Conference (ESWC 2013), volume 7882 of Lecture Notes in Computer Science, pages 382–396. Springer, May 2013.

[108] Nadine Steinmetz and Harald Sack. About the influence of negative context. In Proceedings of 7th International Conference on Semantic Computing (ICSC 2013). IEEE Computer Society, 2013.

[109] Nadine Steinmetz, Magnus Knuth, and Harald Sack. Statistical analyses of named entity disambiguation benchmarks. In Proceedings of 1st International Workshop on NLP & DBpedia at ISWC2013, volume 1064 of CEUR Workshop Proceedings. CEUR-WS.org, 2013.

[110] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In Proceedings of the 16th international conference on World Wide Web, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242667. URL http://doi.acm.org/10.1145/1242572.1242667.

[111] Nina Tahmasebi, Gerhard Gossen, Nattiya Kanhabua, Helge Holzmann, and Thomas Risse. Neer: An unsupervised method for named entity evolution recognition. In Martin Kay and Christian Boitet, editors, Proceedings of the 24th International Conference on Computational Linguistics (Coling 2012), Mumbai, India, December 2012. Coling 2012 Organizing Committee. URL http://L3S.de/neer-coling/.

[112] Peter Tarasewich. Towards a comprehensive model of context for mobile and wireless computing. In AMCIS, page 15. Association for Information Systems, 2003. URL http://dblp.uni-trier.de/db/conf/amcis/amcis2003.html#Tarasewich03.

[113] Emma Tonkin and Marieke Guy. Folksonomies: Tidying up tags? D-Lib, 12(1), January 2006. URL http://www.cs.bris.ac.uk/Publications/pub_info.jsp?id=2000478.

[114] Gerald Töpper, Magnus Knuth, and Harald Sack. Dbpedia ontology enrichment for inconsistency detection. In Proceedings of the 8th International Conference on Semantic Systems, I-SEMANTICS '12, pages 33–40, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1112-0. doi: 10.1145/2362499.2362505. URL http://doi.acm.org/10.1145/2362499.2362505.

[115] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, pages 173–180, Morristown, NJ, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073478. URL http://portal.acm.org/citation.cfm?id=1073445.1073478.

[116] Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. Joint learning improves semantic role labeling. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05, pages 589–596, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219913. URL http://dx.doi.org/10.3115/1219840.1219913.

[117] Ton van der Wouden. Negative Contexts: Collocation, Polarity and Multiple Negation. Taylor & Francis, 1997.

[118] Teun A. van Dijk. Text and context. Explorations in the semantics and pragmatics of discourse. Longman Linguistics Library, London, 1977.

[119] Howard D. Wactlar, Alexander G. Hauptmann, Michael G. Christel, Ricky A. Houghton, and Andreas M. Olligschlaeger. Complementary video and audio analysis for broadcast news archives. Communications of the ACM, 43(2):42–47, February 2000. ISSN 0001-0782. doi: 10.1145/328236.328144. URL http://doi.acm.org/10.1145/328236.328144.

[120] Jörg Waitelonis and Harald Sack. Towards exploratory video search using linked data. Multimedia Tools and Applications, pages 1–28, 2011. ISSN 1380–7501. URL http://dx.doi.org/10.1007/s11042-011-0733-1.

[121] Maciej Witek. Introduction. Philosophica, 75, 2005.

[122] Stephen Wolfram. A quick introduction to wolfram|alpha. http://tinyurl.com/q735x7, May 2009. Last visited on November, 13th 2013.

[123] H-J. Yang, B. Quehl, and H. Sack. Text detection in video images using adaptive edge detection and stroke width verification. In Proc. of 19th Int. Conf. on Systems, Signals and Image Processing (IWSSIP), pages 9 – 12, Vienna, Austria, April 11–13 2012. IEEE Computer Society.

[124] H-J. Yang, B. Quehl, and H. Sack. A skeleton based binarization approach for video text recognition. In Proc. IEEE Int. WS. on Image Analysis for Multimedia Interactive Service, pages 1–4, Dublin, Ireland, May 2012. IEEE Computer Society.

[125] David Yarowsky. Word-sense disambiguation. In Handbook of Natural Language Processing. Marcel Dekker, Inc., 2000.

[126] Mohamed Amir Yosef, Johannes Hoffart, Ilaria Bordino, Marc Spaniol, and Gerhard Weikum. Aida: An online tool for accurate disambiguation of named entities in text and tables. PVLDB, 4(12):1450–1453, 2011. URL `http://dblp.uni-trier.de/db/journals/pvldb/pvldb4.html#YosefHBSW11`.

DECLARATION

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Dissertation, die anderen Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind durch Angaben der Herkunft kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet.

Potsdam, Dezember 2013

Nadine Steinmetz