

A Personal Research Agent for Semantic Knowledge Management
of Scientific Literature

Bahar Sateli

A Thesis
in the Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Computer Science) at
Concordia University
Montréal, Québec, Canada

February 2018

© Bahar Sateli, 2018

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Bahar Sateli**

Entitled: **A Personal Research Agent for Semantic Knowledge Management
of Scientific Literature**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to
originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Georgios Vatisas

_____ External Examiner
Dr. Guy Lapalme

_____ Examiner
Dr. Ferhat Khendek

_____ Examiner
Dr. Volker Haarslev

_____ Examiner
Dr. Juergen Rilling

_____ Supervisor
Dr. René Witte

Approved by _____
Dr. Volker Haarslev, Graduate Program Director

9 April 2018 _____
Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

A Personal Research Agent for Semantic Knowledge Management of Scientific Literature

Bahar Sateli, Ph.D.

Concordia University, 2018

The unprecedented rate of scientific publications is a major threat to the productivity of knowledge workers, who rely on scrutinizing the latest scientific discoveries for their daily tasks. Online digital libraries, academic publishing databases and open access repositories grant access to a plethora of information that can overwhelm a researcher, who is looking to obtain fine-grained knowledge relevant for her task at hand. This overload of information has encouraged researchers from various disciplines to look for new approaches in extracting, organizing, and managing knowledge from the immense amount of available literature in ever-growing repositories.

In this dissertation, we introduce a *Personal Research Agent* that can help scientists in discovering, reading and learning from scientific documents, primarily in the computer science domain. We demonstrate how a confluence of techniques from the Natural Language Processing and Semantic Web domains can construct a semantically-rich knowledge base, based on an inter-connected graph of scholarly artifacts – effectively transforming scientific literature from written content in isolation, into a queryable web of knowledge, suitable for machine interpretation.

The challenges of creating an intelligent research agent are manifold: The agent’s knowledge base, analogous to his *brain*, must contain accurate information about the knowledge ‘stored’ in documents. It also needs to know about its end-users’ tasks and background knowledge. In our work, we present a methodology to extract the rhetorical structure (e.g., claims and contributions) of scholarly documents. We enhance our approach with entity linking techniques that allow us to connect the documents with the Linked Open Data (LOD) cloud, in order to enrich them with additional information from the web of open data. Furthermore, we devise a novel approach for automatic profiling of scholarly users, thereby, enabling the agent to personalize its services, based on a user’s background knowledge and interests. We demonstrate how we can automatically create a semantic vector-based representation of the documents and user profiles and utilize them to efficiently detect similar entities in the knowledge base. Finally, as part of our contributions, we present a complete architecture providing an end-to-end workflow for the agent to exploit the opportunities of linking a formal model of scholarly users and scientific publications.

Acknowledgments

It is indeed the greatest joy and pride in my life to write this acknowledgment for my doctoral dissertation. The overwhelming sense of accomplishment brings warmth to my heart, which I would like to share with those who made it possible for me to embark on this journey and stood by me at every step of the way.

First and foremost, I humbly express my sincere and profound gratitude to my supervisor and mentor, Dr. René Witte, who, for many years, was the guiding light in my graduate studies. This dissertation would have never been possible without his invaluable guidance, insightful feedback, remarkable patience, and meticulous editing. I am forever indebted to him for seeing more in me than I saw in myself. Thank you for teaching me that no dream is ever too big to pursue.

I was tremendously fortunate to conduct parts of my research on scholarly user profiling at the Friedrich Schiller University of Jena in Germany, in collaboration with Felicitas Löffler and Prof. Dr. Birgitta König-Ries. I would like to take this opportunity to thank them for welcoming me to their research group with open arms, as well as their hospitality during my visits. My research benefited immensely from their contributions.

It is also my absolute honour and duty to thank my parents, Parvin and Mostafa, who generously and wholeheartedly gave me their unconditional love and endless support throughout these years. This dissertation pales into insignificance compared to the sacrifices they made for me to make it this far. I am grateful for your trust and confidence in me and for giving me the freedom to pursue my dreams. I, forever, will treasure your love in my heart.

I can not go on without thanking my brother, Babak, who was my source of inspiration to enter the fascinating world of computing. He taught me to write my very first lines of code and I hope I have made him proud. This dissertation is dedicated to him and my parents.

This work is also dedicated to my husband, Mohammad, for his unwavering love and encouragements during the pursuit of my studies. Thank you for always believing in me, for accompanying me on those long nights of writing, and for reminding me to endure during the tough times. I am truly thankful for having you in my life.

Contents

List of Figures	ix
List of Tables	xi
List of Acronyms	xiii
1 Introduction	1
1.1 Motivation	2
1.2 Significance of the Work	3
1.3 Summary of Research Contributions	5
1.4 Outline	5
2 Requirements Analysis	6
2.1 Requirements Analysis	6
2.1.1 Functional Requirements	6
2.1.2 Non-Functional Requirements	8
2.2 Research Goals	9
2.2.1 Goal 1: Design a Semantic Scholarly Knowledge Base	10
2.2.2 Goal 2: Automatic Construction of the Knowledge Base	11
2.2.3 Goal 3: Design of a Personal Research Agent	12
2.3 Summary	13
3 Background	14
3.1 Natural Language Processing	14
3.1.1 Evaluation and Metrics	16
3.2 Vector Space Model	18
3.2.1 The VSM in an Information Retrieval Context	19
3.2.2 Evaluation and Metrics	22

3.3	Linked Data Principles	24
3.3.1	Web of Linked Open Data	24
3.3.2	Evaluation	25
3.4	User Modeling and Recommender Systems	26
3.5	Agents and the Semantic Web	28
3.6	Summary	30
4	Literature Review	31
4.1	Semantic Publishing	31
4.1.1	Scientific Literature Markup	32
4.1.2	Discussion	36
4.2	Argumentation Mining in Scientific Literature	36
4.2.1	Proposed Schemas	36
4.2.2	Manual Approaches for Rhetorical Entity Markup	37
4.2.3	Automatic Approaches for Rhetorical Entity Detection	38
4.2.4	Other Disciplines	39
4.2.5	Discussion	40
4.3	Scholarly Profiling and Recommendation Tools	41
4.3.1	Implicit User Profiling	41
4.3.2	Scientific Literature Recommender Tools	43
4.3.3	Discussion	46
4.4	Summary	46
5	Personal Research Agents Design	47
5.1	An Abstraction Model for the Knowledge Base	47
5.1.1	Competency Question-Based Ontology Construction	47
5.1.2	Domain Model	49
5.1.3	Semantic Mappings	50
5.1.4	The Knowledge Base	53
5.2	Semantic Modeling of Scholarly Literature	53
5.2.1	Bibliographical Metadata	54
5.2.2	Scientific Discourse Modeling	55
5.3	Semantic Modeling of Scholarly Users	61
5.3.1	A Schema for User Knowledge Representation	62
5.3.2	A Schema for Scholarly User Profiles	63
5.4	Semantic Modeling of Personal Research Agents	64

5.5	Summary	66
6	Automatic Knowledge Base Construction	67
6.1	Extraction of Bibliographical Metadata	67
6.1.1	Pre-processing Phase	68
6.1.2	Text Segmentation	69
6.1.3	Detection of Authorship Metadata	69
6.1.4	Detection of References	74
6.2	Extraction of Rhetorical and Named Entities	75
6.2.1	Common Linguistic Patterns in Rhetorical Entities	76
6.2.2	Detection of Domain Concepts as Named Entities	78
6.3	Scholarly User Profile Population	78
6.4	Triplification: Transforming Annotations to Triples	79
6.4.1	Mapping Language	80
6.4.2	URI Generation	83
6.5	Summary	84
7	Implementation	85
7.1	Document Pre-processing Pipeline	85
7.2	Semantic Publishing Pipeline	87
7.2.1	Text Segmentation	87
7.2.2	Authorship Metadata Extraction	89
7.3	Discourse Analysis Pipeline	90
7.3.1	Rhetector: Automatic Detection of Rhetorical Entities	91
7.3.2	LODtagger: Named Entity Detection and Grounding	92
7.4	ScholarLens: Semantic User Profiling Pipeline	93
7.5	Automatic Knowledge Base Population	94
7.5.1	LODeXporter: Flexible Generation of LOD Triples	94
7.5.2	Knowledge Base Population with Document Entities	95
7.5.3	Knowledge Base Population with Semantic User Profiles	95
7.6	An Architecture for Personal Research Agents	96
7.6.1	Vector-based Representation of Scholarly Artifacts	96
7.6.2	Semantic Scholarly Services	97
7.7	Summary	106

8	Evaluation	108
8.1	Semantic Publishing Pipeline Evaluation	108
8.1.1	Gold standard	108
8.1.2	Results	109
8.2	Rhetector Pipeline Evaluation	111
8.2.1	Gold standard	111
8.2.2	Results	113
8.3	Semantic User Profiling User Study	116
8.3.1	User Study: Error Analysis	118
8.3.2	Extended Experiments	119
8.4	Semantic Vectors Evaluation	126
8.4.1	Gold standard	126
8.4.2	Experiment Design	126
8.4.3	Results	128
8.5	Summary	129
9	Conclusions	132
9.1	Summary	132
9.2	Future Work	135
	Bibliography	138
	Author's Publications	150
	Appendix A Supplementary Materials	155
	Appendix B ANNIE Part-of-Speech Tagset	157
	Appendix C Referenced Ontologies	159
	Appendix D Example Competency Questions	160
	Appendix E Rhetorical Analysis Resources	161
	Appendix F LODeXporter Mapping File	163
	Appendix G Semantic Vectors Evaluation Results	166
	Appendix H Solr Configuration Schema	169

List of Figures

1	The personal research agent conceptual map	4
2	The GATE Developer environment	16
3	Representation of vectors in the vector space model	19
4	Representation of document vectors in a 3-dimensional space	21
5	Topology of the web of Linked Open Data (LOD) in 2017	26
6	Human-readable version of a DBpedia entity	27
7	The semantic web stack (proposed by Tim Berners-Lee)	29
8	A flexible workflow for scholarly knowledge base construction	48
9	The agent’s knowledge base domain model	51
10	Example processing using our workflow with its input and output	53
11	The agent’s semantic model of bibliographical entities in a document	56
12	Agent’s model of relations between a document and a rhetorical entity	59
13	Agent’s model of named entities in a document	61
14	An RDF graph representing a semantic user profile	64
15	Example literature review task modeling using the agent’s task model	66
16	Automatic segmentation of a scholarly document	70
17	Example rules declaring how NLP annotations should be mapped to semantic triples	82
18	Anatomy of a generated URI for an Author annotation	84
19	The sequence of processing resources in the pre-processing pipeline	88
20	The sequence of processing resources in the Rhetector pipeline	91
21	JAPE rules to extract a Contribution sentence and the generated annotations in GATE	92
22	The sequence of processing resources in the LODtagger pipeline	93
23	A JSON example response from Spotlight and the generated annotation in GATE .	93
24	Annotations for an author, a competence topic, and the generated competency record	94

25	The sequence of processing resources in the LODEXporter pipeline	95
26	An excerpt of the Solr schema to construct semantic vectors	97
27	The complete architecture showing the end-to-end workflow for KB construction . .	98
28	Query to find all Claims and Contributions within a document	99
29	Example entry from the agent’s output in the summary generation task	100
30	Query to retrieve all documents with a contribution related to a topic	101
31	The agent’s output for assisting a researcher in a literature review task	102
32	The agent’s output in recommending related work to a user	104
33	Query to provide learning content for topics new to researcher R_1	105
34	The agent’s output assisting a researcher in understanding unknown topics	106
35	The agent’s output providing an overview of a corpus	106
36	Query to find documents with a novel combination of topics for researcher R_1	107
37	The agent’s output in issuing an alert on discovering new knowledge	107
38	Example query from the semantic publishing challenge and our query results	111
39	An automatically generated user profile in \LaTeX format	117
40	An automatically generated web-based survey using <i>LimeSurvey</i>	120
41	Plots showing the distribution of top-50 competence in full-text and RE-only profiles	125
42	Best performing configuration for document recommendation ($df = 6$)	129
43	The Zeeva wiki user interface	136

List of Tables

1	User stories describing requirements of the agent’s end-users	7
2	Mapping of user groups, their requirements and our research goals	9
3	Information retrieval systems evaluation contingency table	23
4	Overview of existing schemas for rhetorical blocks in scientific literature	37
5	Archetypes of the knowledge base competency questions	49
6	High-level semantic mapping of our PUBO domain model	52
7	Bibliographical metadata vocabularies in our PUBO schema	55
8	High-level semantic mapping of our PUBO argumentation model	58
9	Scholarly user profiles features	62
10	Selected linked open vocabularies for semantic scholar profiles	64
11	Selected linked open vocabularies for our agent’s workflow	65
12	Example annotations generated from the pre-processing pipeline	87
13	Statistics of the semantic publishing challenge gold standard corpora	110
14	Evaluation results of our pipeline on the Semantic Publishing Challenge gold standard	112
15	Evaluation results of the semantic publishing challenge 2016 queries	113
16	Evaluation results of the semantic publishing challenge 2015 queries	114
17	Statistics of the discourse analysis gold standard corpora	114
18	Results of the intrinsic evaluation of Rhetector	115
19	Quantitative analysis of the populated knowledge base with REs and NEs	116
20	Evaluation results for the generated user profiles in the first user study	117
21	Error analysis of the irrelevant competence entries for the pilot study participants .	119
22	Analysis of the survey responses for profiles generated from Full-text and RE Zones	122
23	Precision of full-text profiles with a relevance threshold of Irrelevant (0) & General (1)	123
24	Precision of RE-only profiles with a relevance threshold of Irrelevant (0) & General (1)	124

25	Summary of the scholarly user profiling evaluations	126
26	Example entry from the recommendation gold standard dataset	127
27	Average precision for various recommendation configurations	130
28	Average nDCG for various recommendation configurations	131
29	Mapping of our research goals with their resulting contributions and publications . .	134

List of Acronyms

CSV Comma Separated Values

CQ Competency Question

DCG Discounted Cumulative Gain

GATE General Architecture for Text Engineering

HTML HyperText Markup Language

IR Information Retrieval

JAPE Java Annotation Pattern Engine

JSON JavaScript Object Notation

KB Knowledge Base

LR Language Resource

LOD Linked Open Data

LOV Linked Open Vocabulary

MAP Mean Average Precision

MLT More Like This

nDCG Normalized DCG

NE Named Entity

NER Named Entity Recognition

NLP Natural Language Processing

OCR Optical Character Recognition

OWL Web Ontology Language

PDF Portable Document Format

POS Part-of-Speech

PR Processing Resource

RDF Resource Description Framework

RDFS RDF Schema

RE Rhetorical Entity

SVM Support Vector Machines

SPARQL SPARQL Protocol and RDF Query Language

URI Uniform Resource Identifier

URL Uniform Resource Locator

VSM Vector Space Model

W3C World Wide Web Consortium

XML eXtensible Markup Language

Chapter 1

Introduction

With the increasing rate of scientific output currently being published in ever-growing online repositories, a persisting issue is the management and analysis of the wealth of knowledge contained in scientific articles. In the midst of this deluge of information, one should ask, how much support do researchers have in their research-related tasks? Take, for example, a scientist that has to read up on existing literature relevant to her topic of interest. A keyword-based search in any online digital library or indexing engine, like Google Scholar,¹ typically retrieves thousands of articles in a matter of seconds. The researcher is then left unassisted in triage and curation of the retrieved articles. The same argument can be made when writing scientific literature, like journal manuscripts: There are no existing automatic approaches that can support researchers in comparing their claims against existing work or ensuring that they have fully covered related research in a literature survey. With the current pace of scientific publishing, researchers are always on the verge of being outdated or producing redundant research.

In a commentary for the *Nature* journal in 2001, Tim Berners-Lee et al. predicted that the new semantic web technologies “*may change the way scientific knowledge is produced and shared*” [BLH01].² They envisioned the concept of “*machine-understandable documents*”, where machine-readable metadata is added to articles in order to explicitly mark up the data, experiments and rhetorical elements in their raw text. One major obstacle towards this goal, however, is the fact that for the past hundreds of years, researchers have been using the same medium for the dissemination of information: writing articles in natural languages, designed and published primarily for human reading. Eventually, this intuitive choice caused the wealth of knowledge produced for hundreds of years to be buried deep in disparate libraries, inaccessible to machines for interpretation and simply

¹Google Scholar, <http://scholar.google.ca/>

²Please note that references in abbreviated format, such as [BLH01], can be found in the Bibliography chapter, on page 138, whereas citations to our own publications, like [8], use numerical format and can be found in the Author’s Publications chapter on page 150.

overwhelming for humans to manually curate them.

Two influential articles in 2009 sparked a new series of academic and commercial initiatives to look into research and development of innovative ways of enhanced scientific publishing, now referred to as *Semantic Publishing*. David Shotton – a pioneer in this area – called semantic publishing “*a revolution in scientific journal publishing*” [Sho09]. He took the idea of augmenting articles with metadata further, by envisioning the development of value-added services to facilitate automated ‘understanding’ of articles. These services would integrate information from multiple papers or external sources using Semantic Web technologies. Shotton also predicted that in the next decade, the value of raw text in scientific publishing will decrease, while the development of services and standards in publishing will be an increasing trend that benefits both researchers and businesses. Along the same lines, Attwood et al. [AKM⁺09] published an article in the Semantic Biochemical Journal, calling for an “*international rescue of knowledge*” that is being “*buried in the unmanageable volumes of scientific data*”. They encouraged multi-disciplinary collaborations between life and computer scientists, publishers, database curators and librarians, among others. The advisory of prominent academic figures, together with the exciting business opportunities [GB10] that a web of machine-readable documents would bring to digital publishers, resulted in semantic publishing research and development to gain rapid momentum in recent years.

1.1 Motivation

For the moment, let us assume that the body of knowledge in existing scientific literature were available as machine-readable data. We can envisage intelligent research *agents* that can provide a myriad of services to researchers by exploiting such formalized knowledge: The retrieval of scientific literature will no longer be based on keywords, but on their *contributions* and their semantic relationships, like the similarity of their *claims*. The agents can help researchers by creating automatic summaries of one or multiple documents to bootstrap a literature review task. Young researchers can be supported in learning a topic by a personalized reading assistant that suggests papers to them, based on their background knowledge. Peer reviewers can have agents automatically checking submitted manuscripts for plagiarism and the novelty of their claims against existing literature. Ultimately, the agents can help researchers to find the gaps in existing knowledge and form hypotheses through semantic inferencing, thereby, cultivating literature-based knowledge discoveries.

This dissertation aspires to bring this ambitious vision closer to reality, by creating a **Personal Research Agent** that helps users in research-related tasks, in particular, finding, reading, writing and learning from scientific articles. We populate the agent’s knowledge base, analogous to his *brain*, with information available in a given domain’s literature. This process is carried out in

four incremental steps: (i) We use Natural Language Processing (NLP) techniques to automatically extract key information from articles; (ii) We formalize all the detected entities using W3C³ standard semantic web technologies in order to facilitate their machine-consumption and integration with the web of Linked Open Data (LOD); (iii) We introduce a methodology to represent the agent’s knowledge about users, their tasks and interests, as well as the detected information in the first two steps, based on linked open vocabularies; and (iv) propose a semantic model of the agent’s working context and tasks. Thereby, we enable the research agent to offer *personalized services* on a user-level and task-specific basis, by integrating the information available in user profiles, as well as its knowledge base.

1.2 Significance of the Work

Despite the web’s early-day presumption that a complete index of the scholarly publishing landscape would facilitate finding relevant information, the unprecedented rate of scientific output has rendered large-scale indexing solutions ineffective: Even for trained users who can formulate sophisticated search queries, triaging the enormous amount of results is still a time-consuming task that does not contribute directly to their research activities. On the other end of the spectrum, a multitude of domain-specific, application-oriented tools have been developed to address specific needs of researchers like finding, reading, annotating or organizing scholarly articles [KB15]. However, none of these existing solutions offer a personalized environment that can ‘cut through the clutter’ and actively offer fine-grained access to pertinent knowledge contained within scholarly artifacts.

The notion of a personal research agent, conceptualized in Figure 1, is derived from researchers’ urgent need in dealing with the influx of scientific resources available on the Web, combined with a lack of existing solutions that can fulfill all aspects of a scholar’s workflow. Several user surveys have been conducted in related works [CYY14, NHA09, FGC⁺08], with the aim of eliciting the habits of researchers in interacting with scholarly literature, to identify hindering factors and potential improvement points. Based on the recurring demographics in these studies, we identified the following end-user groups, who can benefit from our personal research agent:

Post-secondary students, in particular graduate students and post-doctoral fellows. This group represents users, ranging from novice to experienced readers, looking for assistance in understanding an article’s content or finding relevant work for their own research topic. This user group also tends to demonstrate an “*anomalous knowledge state*” [BOB82], where they are unable to precisely formulate their information needs due to unfamiliarity with the new domain’s concepts and terminologies.

³World Wide Web Consortium (W3C), <http://www.w3.org/>

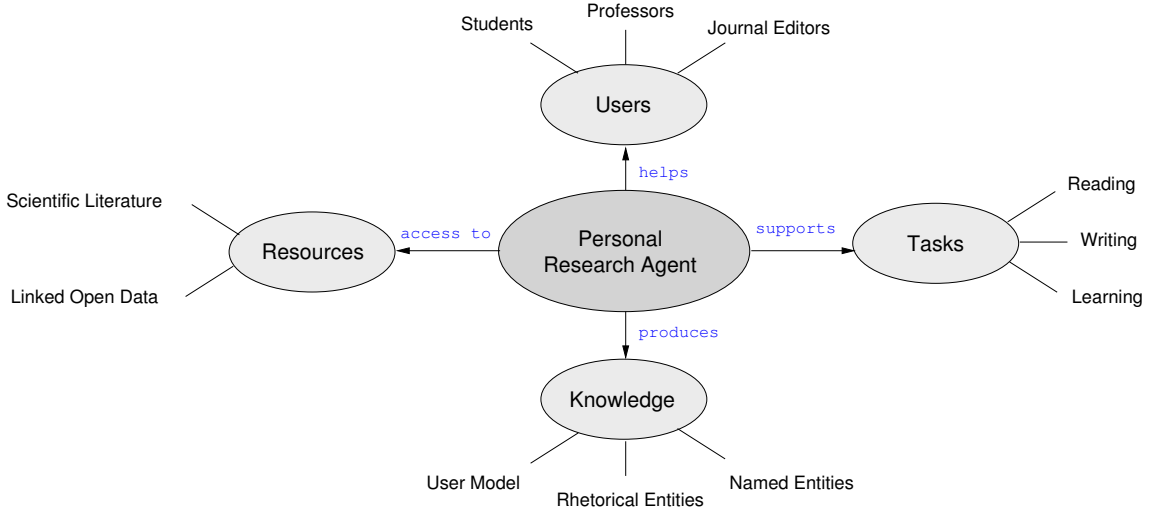


Figure 1: The personal research agent conceptual map

Researchers, including junior and senior researchers, professors, as well as practitioners in research and development positions. These users are typically more experienced than the former group and have a broader range of background knowledge. However, users in this group are specialized in a narrow field and are typically interested in discovering advances in their designated area.

We also include a new user group with the agent’s stakeholders, as substantiated by recent and on-going industrial efforts (see Section 4.1) from communities involved in the semantic publishing domain:

Scientific publishing bodies, including users involved in publication and dissemination of scholarly output, such as journal editors, and publishing companies. Users in this group are interested in obtaining a high-level view of the scientific output landscape for publishing, editing, reviewing or decision-making purposes.

Evidently, constructing a comprehensive solution, such that it can fulfill the information needs of the above user groups, is a non-trivial task. In this dissertation, we argue that we can automatically construct a *knowledge base* of scholarly artifacts to empower research agents proactively assisting end-users in a variety of scientific tasks. Hence, the distinguishing feature of our agent lies within the underlying design and technologies used to build its knowledge base – an inter-connected registry of information that holds a semantically-rich model with machine-readable metadata. Furthermore, in lieu of creating multiple ad-hoc tools to partially comprise end-users’ tasks, our research agents offer an adaptable information modeling methodology that provides for a dynamic formation of diverse and novel services through formulating complex queries and connecting them with the immense amount of external knowledge available on the Web.

1.3 Summary of Research Contributions

The overall goal of this dissertation is to create a personal research agent that can help scholarly users in their research-related tasks, with a primary focus on the computer science (informatics) domain. Towards this end, we studied the current problems researchers are faced with when working with scientific literature. We gathered various information needs from available relevant surveys and translated them to a set of requirements. Towards the realization of the agent and its services, our contributions can be summarized as follows:

- An open, interoperable design for semantic modeling of scientific literature and their authors using Semantic Web techniques [6, 11, 22, 21].
- Investigating automated approaches for fine-grained structural and semantical analysis of scientific text in the computer science domain [21], as well as profiling of scholarly users [10, 11]
- Development of multiple text mining pipelines for user and document modeling, released now as open-source software [20, 23].
- Development of a novel, flexible methodology to transform literature from textual content into queryable semantic graphs [29].
- Scalable, automated population of a semantic knowledge-base from large collections of domain publications [24, 29].
- The first design of a personal research agent and its scholarly services [25, 26].

A complete list of published works relevant to this dissertation can be found in the author’s publication section on page 150. Please refer to Appendix A for the supplementary materials related to this dissertation.

1.4 Outline

This dissertation is structured as follows: In the next chapter, we will lay out our research goals and the respective methodologies used in our approach. Chapter 3 covers the foundations related to this work, in particular the text mining and semantic web technologies used for knowledge base construction, as well as the metrics used to evaluate them. Chapter 4 covers existing works relevant to each of our agent’s design aspects. Chapters 5, 6 and 7 have a similar structure, whereby each of our agent’s design goals is provided as an abstract design, an automated approach, followed by its implementation details. We evaluate our agent’s goals in Chapter 8 and conclude the dissertation in Chapter 9.

Chapter 2

Requirements Analysis

In this chapter, we elaborate the contributions of this dissertation as briefly outlined in Chapter 1. We will then lay out our research goals, their corresponding requirements and the methodology used to fulfill them.

2.1 Requirements Analysis

Inspired by the agile approach in developing information systems, we gathered the requirements of our agent’s design in form of *user stories* [Coh04] – short descriptions of features from the perspective of end-users, as well as the system development. Table 1 shows the user stories we constructed from the users’ survey responses published in [CYY14, NHA09, FGC⁺08]. Note that, where applicable, we combined multiple user stories into one requirement.

2.1.1 Functional Requirements

Functional requirements dictate the services a research agent must provide to its end-users. Note that the list of possible services is by no means limited to the ones described here; Rather, the following requirements are pertinent to the surveys mentioned above:

Requirement #1: View a Summary of an Article. Abstracts of scholarly articles are written in a generic style that rarely contains any of the authors’ scientific arguments. To help users in triage and comparing scientific articles, they should have access to a summary of an article’s contributions prior to reading its full-text.

Requirement #2: Obtain Support in Literature Review. Writing a comprehensive literature review is an infamously time-consuming task for researchers. End-users should be able to

Table 1: User stories describing requirements of the agent’s end-users

As a . . .	I want to . . .	so that . . .
student	find existing works related to my research topic	I can write about them in my literature review chapter
student	know how a certain methodology is used in a specific field of computer science	I can learn about how this methodology is applied to solve a problem
student	view a definition of all the terms in an article that I do not know about	I can better understand the article’s meaning
researcher	read a summary of the most important parts of an article	I can save time in judging its relevance to my work
researcher	compare my work against existing publications	I can better position my research work against others
researcher	be alerted about new knowledge published regarding my research interests	I can keep my knowledge up-to-date
researcher	gather information from published works in a journal or a conference proceedings	I can find new ideas for my research from the trending topics
journal editor	obtain an overall view of the authors who published in my journal and a summary of their contributions	I save time in writing the editor’s note
journal editor	find a prolific author who is competent on a certain topic	I can assign him a peer review task for my journal
funding agency	obtain a visualization of the current research workers and topic trends	I can better allocate my budget

specify an author or a topic (or both) and receive all relevant documents and their contributions. The user can then examine the contributions and write a comprehensive natural language description for her task.

Requirement #3: Find Related Work. End-users need to automatically find other scholarly works related to their research interests, so that they can compare their contributions against existing research.

Requirement #4: Learn about a Topic from the Literature. End-users, like graduate students and interdisciplinary researchers, need assistance in understanding literature, particularly when encountering a topic they have not seen before. The agent must be able to detect *new* topics in a document, with respect to the background knowledge of its end-user, and help her understand the topic through integration of external information.

Requirement #5: View Contributions of an Author, Research Group or Conference.

End-users need to be able to specify an author, group, or conference and receive a high-level view of their contributions, i.e., their published work metadata, as well as a summary of their research contributions.

Requirement #6: Discover Relevant New Knowledge. End-users must be able to receive alerts from the agent only when new knowledge relevant to their research interests becomes available. This new service should only inform the user when a new publication becomes available in the knowledge base, which has a novel combination of topics that the user has not seen before.

2.1.2 Non-Functional Requirements

The construction of a personal research agent also mandates a set of design constraints, such that not only it can fulfill its end-users' requirements, but also the system's as a whole:

Requirement #7: An Interoperable, Open Design. The functional requirements enumerated above are just but a few envisioned services that an intelligent research agent can offer. A multitude of other novel services can be developed to exploit the knowledge amassed from scholarly artifacts. To facilitate the creation of such services, without the need to develop multiple specialized applications, the agent's design must be based on interoperable, open, standards-based knowledge representation techniques.

Requirement #8: Explicit Semantics. In view of a decentralized architecture for intelligent research agents and their knowledge exchange with other scholarly tools (or agents), the agent's knowledge model must be enriched with machine-readable semantics, so that it can be unambiguously exchanged, utilized and integrated with other resources.

Requirement #9: Personalized Services. The ever-growing rate of scientific publishing exceeds the human ability to keep up with the latest discoveries. Therefore, the agent must be able to proactively look for relevant knowledge from available literature, based on the end-users' tasks and interests.

Requirement #10: Scalability. The increasing rate of scholarly publications is a sign of the dramatic growth in disseminated knowledge. The agent's design must be able to handle the extraction, storage and application of information required to fulfill a user's task in its knowledge base in a scalable manner.

Table 2: Mapping of user groups, their requirements and our research goals

Req.	Goal 1			Goal 2	Goal 3	User Groups
	Goal 1.1	Goal 1.2	Goal 1.3			
R1	✓	✓	–	✓	–	All groups
R2	✓	✓	–	–	–	Researchers, Students
R3	✓	✓	✓	–	✓	Researchers, Students
R4	✓	✓	✓	✓	✓	Students
R5	✓	✓	✓	✓	✓	Publishers, Students
R6	✓	✓	✓	✓	✓	All groups
R7	✓	✓	✓	✓	–	n/a
R8	✓	✓	✓	✓	–	n/a
R9	–	–	–	–	✓	n/a
R10	–	–	–	✓	–	n/a

2.2 Research Goals

In this section, we break down the overall vision of this dissertation into smaller, manageable research goals, so that they can be individually defined and evaluated. These goals are highly cohesive and complementary, in the sense that they are incrementally built upon each other’s output, attaining the ultimate goal of a formal representation of scientific users and artifacts. Table 2 provides a mapping between the requirements stated above and our research goals to fulfill them.

Satisfying the requirements described above necessitates a solution that goes beyond bibliographical management and retrieval of scholarly documents. Therefore, a significant contribution of this dissertation is the progressive construction of a scholarly knowledge base, based on W3C recommended open technologies (Requirement #7), as opposed to creating large-scale indexes or single-task applications. A multitude of services can then be developed based on this rich repository of machine-readable knowledge available to our agent for a given task. By combining intelligent agent design principles [Cao15] with the power of knowledge based-systems [DL82], the agent can offer numerous services to help researchers in conducting complex tasks, which we categorize into three cross-cutting scholarly use cases:

Discovery. Contrary to most search engines, where the relevance of a document towards an information need is a function of query term frequency and matching bibliographical metadata, the agent can help users in *knowledge discovery*, by presenting end-users with literature that

are *semantically* related to the user’s information needs. That is, the agent is capable of examining the (full-text) content of each document to determine its pertinence and importance for a given task, like summarization. Such an approach differs from conventional information retrieval techniques and is crucial in understanding complex scholarly texts with a wide range of vocabularies.

Navigation. Current document management systems offer a navigation of documents based on bibliographical metadata, such as authors, affiliations and references. Our agent is able to recommend documents for its end-users based on the similarity of their arguments, like the contributions. Such an overview of the scientific landscape can also help end-users wishing to grasp a panned view of the publishing landscape, such as funding agencies or publishing companies.

Understanding. As a *personal* research agent, our approach acknowledges its end-users’ varying levels of competency. For example, it can distinguish between a graduate student, who is learning a new topic, from a professor looking for the latest discoveries in a narrow field. The agent provides learning assistance in a non-intrusive way, e.g., a student will be able to receive learning aid in context as he is reading a new article, but the professor, with a comprehensive background knowledge model, will only receive alerts about new knowledge found in a designated domain.

2.2.1 Goal 1: Design a Semantic Scholarly Knowledge Base

Our first research goal is to design a semantic model for the agent’s knowledge base of scholarly artifacts. The knowledge base contains metadata about scholarly literature, its end-users, and relevant information extracted from their context.

Goal 1.1: A Semantic Model for Scholarly Bibliographical Metadata

Fulfilling the idea of a personal research agent is substantially determined by the approach to populate its knowledge base. In our approach, the agent’s knowledge base in part includes metadata extracted from scholarly literature. We aim at extracting bibliographical metadata from scholarly articles, which the agent will need for organization and navigation within and across documents. We also dissect the full-text of a document into several meaningful sections, and further analyze each article for its categorical elements, like title, authors, affiliations, keywords, sections and references. For example, finding and showing all papers written by an author (Requirement #5) or listing the metadata of a similar document within a related work task (Requirement #3), requires the agent to store and manage information, such as title and authorship metadata, to uniquely identify the

relevant artifacts in its knowledge base. Where available, the agent can cross-reference its knowledge base with existing online bibliographical repositories.

Goal 1.2: A Semantic Model for Scholarly Argumentations

Many user stories, such as summarization (Requirement #1) or finding related work (Requirement #3), require that the agent gathers an understanding about the contributions of an article, so that it can compare it against the interests of a user, a given document, or the user’s manuscript in a writing task (Requirement #2). In our approach, we model the meaning of a scholarly document as a collective set of sentences and domain topics mentioned in the full-text of documents. Such a semantic model can also be inter-linked with external knowledge contained inside other documents in the knowledge base, as well as the wealth of information available on the web of Linked Open Data.

Goal 1.3: A Semantic Model for Scholarly Users

The end-user groups described in Section 1.2 clearly have different knowledge levels, tasks and interests. The agent also dedicates parts of its knowledge base for modeling its end-users. The goal here is to construct a *user profile* for each individual end-user in the agent’s knowledge base, which is gradually revised as more information about the user’s context becomes available. This user model is then taken into account when offering assistance to the user, hence, providing a personalized experience in working with the intelligent agent, for instance, in issuing alerts (Requirements #4 and #6). Each user profile contains two types of information:

Contextual Information. Scholars often work in a social context. Most end-users are affiliated with an academic organization, might have published their own articles or have read a set of articles written by others. The knowledge base maintains the contextual information about end-users, as well as all authors extracted from documents (Requirements #3 and #5).

Background Knowledge. Proactively finding potentially interesting or relevant literature for end-users (Requirement #6) requires the agent to maintain a formal model of each user’s background knowledge. Our goal here is to design a formal, expressive model of what topics a user is competent in or interested about, such that it can be compared against the knowledge extracted from documents.

2.2.2 Goal 2: Automatic Construction of the Knowledge Base

The immense amount of available literature and their unprecedented growth rate cripples any manual effort in populating the agent’s knowledge base with relevant information. Additionally, the

complex discourse models typically found in scholarly literature makes the automatic construction of a knowledge base a complex task for the agent.

A deep understanding of natural language documents is an intricate task and also an open research question in the NLP domain. To sidestep this issue, we show how the semantic document model conceived in our first goal can be populated without a complete understanding of their full-text content. We show that the agent can segment the main matter of each document into several rhetorical blocks, shown to be the most interesting parts to human readers [NHA08] and needed for tasks like generating an extractive summary (Requirement #1).

Furthermore, manually curating, storing and maintaining the entire established knowledge in an area, such as its nomenclature, in the knowledge base, is a naïve proposition and impractical. Therefore, we investigate inter-connecting the content of scientific literature to existing open knowledge sources on the Web, where possible.

We also address the problem of the *automatic* generation of semantic scholarly profiles. One approach to populate a user model is to ask the user about all the topics she knows and the papers she has read. Such an obtrusive approach, however, would be tedious for users; e.g., a professor might know hundreds of computer science-related concepts. We investigated a new approach, where the user model is first populated by inquiring about the reading and writing history of a user in order to bootstrap the agent’s user knowledge model. Subsequently, as the user interacts with the research agent, she can give both implicit and explicit feedback. This way, the agent can model what the user knows, without her manually specifying the entities.

2.2.3 Goal 3: Design of a Personal Research Agent

The last goal of this dissertation is a semantic design for a personal research agent. Inspired by our requirements analysis (see Section 2.1) and the available information in the agent’s knowledge base, we devised a set of semantic scholarly services that the agent can perform to fulfill its users’ information needs. We investigated how each service can be broken down into a set of *tasks* that can exploit the agent’s knowledge base and integrate additional information from the Linked Open Data cloud. Additionally, we introduce a novel semantic vocabulary for the description of personal research agents and their working context, including the artifacts they can process or generate for an end-user, using W3C recommended frameworks. Finally, we implement the agent’s services as a set of graph-based queries over the knowledge base structure and show sample outputs. We leave the concrete implementation of the agent’s graphical user interface outside the scope of this work, but provide the implementation details of the agent’s services as pseudo-code, where applicable.

2.3 Summary

This chapter provided an overview of this dissertation's contributions. We iterated three complementary research goals and illustrated how each one will contribute to the overall aim of constructing intelligent, personal research assistants. In the next chapter, we provide a brief foundation of the text analysis and semantic web techniques used in our approach.

Chapter 3

Background

The work presented in this dissertation combines various practices from multiple fields of study in computer science. In this chapter, we provide a brief foundation of the underlying techniques used in our research. If you are already familiar with these concepts, you can safely move on to the next chapter, as cross-references are provided wherever specific background information is required.

3.1 Natural Language Processing

Natural Language Processing (NLP) is a branch of computer science that uses specific techniques from the Artificial Intelligence and Computational Linguistics domains to process textual content written in natural languages, like English. NLP is a broad term encompassing a variety of analysis routines, like text segmentation, as well as domain-specific applications, such as question-answering systems. One popular application from the NLP domain is *text mining*. Text mining aims at deriving meaningful structured information from usually free-form text and representing it in a (semi-)structured format. Text mining has several subtasks that analyze text at different levels of abstraction. Some analysis tasks process the syntactical features of a text, like finding the grammatical category of each word in a text, while others work on labeling spans of text with pre-defined categories, such as names of persons, companies or locations. Most text mining techniques are performed using a pattern-matching or statistical approach, or a combination of both.

As using NLP techniques gained popularity in the software development community, several frameworks were implemented that allow language engineers to create reusable text processing components with elemental programming skills, while at the same time enabling software developers to integrate text analytics capabilities within their applications, without the need to have a substantial linguistics background. The Apache UIMA¹ and the General Architecture for Text Engineering

¹Unstructured Information Management Architecture (UIMA), <https://uima.apache.org>

(GATE)² [CMB⁺11] framework are among the most widely-used, open source NLP development platforms. These frameworks offer a suite of tools and off-the-shelf components that can be customized or combined to build concrete text mining applications.

In this dissertation, we extensively used the GATE framework version 8.4 to develop text mining solutions, specifically designed to process scientific literature. We reused some of its readily available components, particularly for the pre-processing phase. In addition, we developed custom components and plugins for the GATE environment using its libraries (referred to as GATE Embedded), available as a set of Java archives (JARs) under the GNU Lesser General Public Licence 3.0³ license. We provide a brief introduction to the GATE architecture and its associated terminology in this section.

GATE provides an extensible architecture, where arbitrary text processing capabilities can be added to it through the implementation of *plugins*. Plugins are comprised of one or more components (referred to as *resources*) that can represent textual data (e.g., documents, corpora, or lexicon sets) or algorithmic modifications of said data. In GATE terminology, the former are referred to as *Language Resources* (LRs) and the latter are known as *Processing Resources* (PRs). Language engineers can reuse existing processing resources or develop custom ones based on the GATE library and specify an order (or a set of conditions) for their sequential execution. An array of processing resources can be bound together to form a GATE *pipeline* that executes over a collection of documents (i.e., a *corpus*). Typically, when a pipeline is run over a given corpus, the full-text of each document is provided to the first processing resource in the sequence and subsequently handed down the stream of remaining PRs. Each PR may produce additional metadata on the text in form of an *annotation* or modify an existing annotation on the document. Further information about an annotation is stored as its *features*, internally represented as a map of key-value pairs with arbitrary datatypes.

GATE also provides a pattern-matching engine called JAPE [CMT00] (Java Annotation Pattern Engine)⁴ that allows developers to perform regular expressions over annotations in a document. The patterns are internally transformed into finite-state transducers and executed over the graph-based structure of GATE documents. For every instance of a matching sequence in a text, a user-defined action can be conducted on the underlying span of characters, for instance, adding or removing an annotation or any arbitrary computation implemented using the Java programming language.

The two GATE plugins used in this dissertation are ANNIE [CMBT02] and Tools. GATE’s ANNIE plugin is a bundle of processing resources, available as part of the GATE public distribution. ANNIE can perform basic pre-processing steps on English text, such as breaking down a text into sentences and further processing them for semantic entities, like names of persons or countries, using a custom dictionary and hand-written pattern-matching rules. The Tools plugin is a collection of

²General Architecture for Text Engineering (GATE), <http://gate.ac.uk>

³GNU Lesser General Public Licence 3.0, <https://www.gnu.org/licenses/lgpl-3.0.html>

⁴JAPE, <https://gate.ac.uk/sale/tao/splitch8.html>

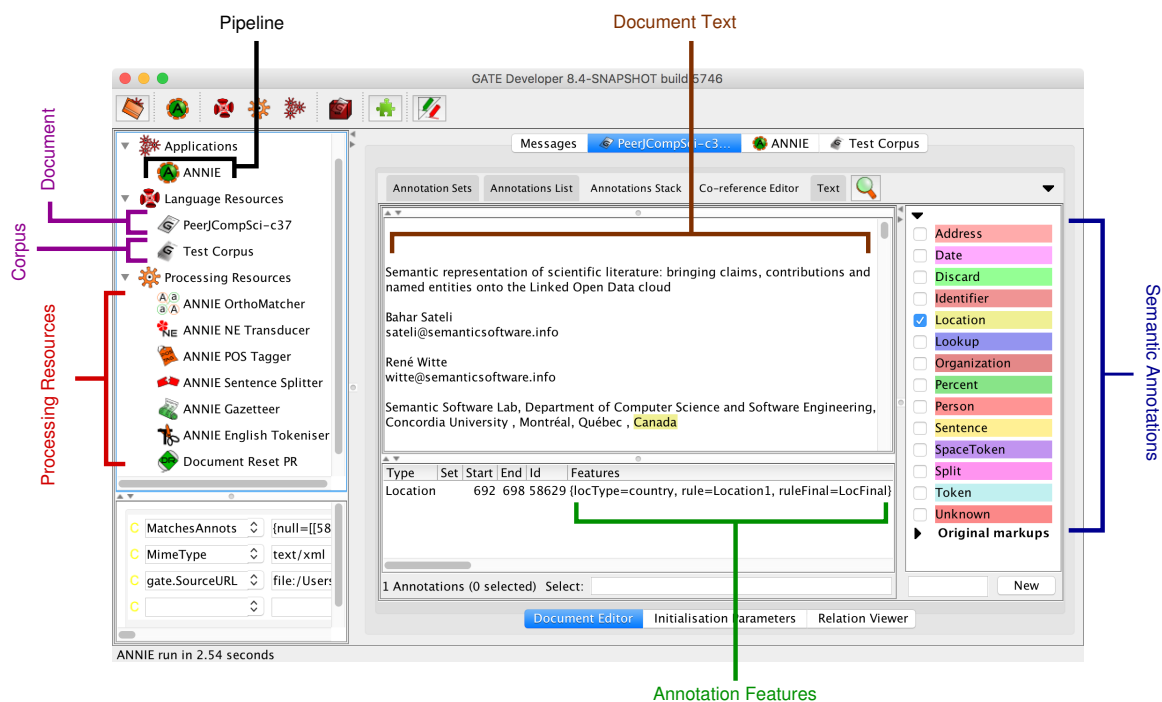


Figure 2: The GATE Developer environment, showing the ANNIE processing resources and the generated semantic annotations

useful auxiliary processing resources, such as PRs for moving annotations within sets in a document or across documents in a corpus. Figure 2 shows the GATE *Developer* environment and the colour-coded annotations generated from executing the ANNIE pipeline over a sample document.

3.1.1 Evaluation and Metrics

The evaluation of NLP applications is conducted in two fashions: In an *intrinsic* study, the output of a text mining system is compared against a set of pre-defined correct answers and the system’s agreement with the ground truth is quantified. In contrast, in an *extrinsic* approach, the impact of an NLP system ‘as a whole’ is measured within the context of a user study, for instance, its effectiveness for a given task. While extrinsic evaluations are complicated to design and reproduce, intrinsic evaluations are done objectively against a so-called *gold standard*. Gold standard corpora contain sets of documents, manually annotated by one or more human experts with the correct results, expected to be generated by a text mining system. For instance, GATE provides a Corpus Quality Assurance plugin that can compare annotations generated by a pipeline against a ‘*goldstandard*’ annotation set. The plugin then calculates whether the overlapping annotations in both sets have (i) the correct span in text (start and end offsets), (ii) the same semantic type, and (iii) the same feature map. Thereby, by comparing the two annotation sets, it can calculate the following numbers:

True Positive (TP) is the number of annotations that match in both sets in terms of their offsets in text, semantic type and features.

False Positive (FP) is the number of annotations incorrectly identified by the pipeline in a text. FPs are calculated by examining the set difference between the pipeline output and the goldstandard annotations.

True Negative (TN) is the number of annotations that are not present in either of the annotation sets. That is, these are spans of text correctly identified by the pipeline to have no annotations.

False Negative (FN) is the number of annotations missed by the pipeline in a text.

Once these numbers are calculated, three standard metrics are reported: *Precision* (P) is the fraction of correct annotations generated by the pipeline, whereas *Recall* (R) is the fraction of detected annotations over all annotations in the goldstandard set. Precision and recall are calculated based on the formulas in Equations 1 and 2, respectively:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

Since the criteria for matching annotations depend on both their offsets and the semantic label, partially correct annotations (i.e., annotations that have the correct semantic type but different character offsets in a text) should also be taken into account during P and R calculations. Conventionally, each metric is computed in three modes: (i) a ‘*strict*’ mode that considers all partially correct responses as incorrect, (ii) a ‘*lenient*’ mode, which treats all partially correct responses as correct, and (iii) an ‘*average*’ mode that allocates a half weight to partially correct responses.

There is an inherent trade-off between precision and recall. Recall is a non-decreasing function of the number of detected annotations, while precision usually decreases as the number of detected annotations goes up. Therefore, in designing NLP systems emphasis is given to one of the two metrics but performance reports include both measures. Alternatively, the *F-measure* is used as a single metric to report the system’s effectiveness, which calculates a weighted harmonic mean of the two aforementioned metrics, as shown in Equation 3 [MRS08]:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2P + R} \quad (3)$$

where β is used to emphasize either precision or recall in calculating the F-measure. In a *balanced F-measure*, equal weights are given to precision and recall, and the metric is commonly referred to as F_1 -measure, as shown in Equation 4:

$$F_1 = \frac{2P \cdot R}{P + R} \quad (4)$$

Metrics like Precision, Recall and F-Measure can be reported based on a ‘*micro*’ and ‘*macro*’ summary. Micro averaging treats the entire corpus as one large document. Correct, spurious and missing annotations are calculated from the entire corpus, and P, R, and F are calculated accordingly. Macro averaging, however, calculates P, R and F on a per-document basis, and then averages the results.

3.2 Vector Space Model

The Vector Space Model (VSM) plays an essential role in the contributions of this dissertation, and therefore, we provide a fundamental understanding of its principles in this section. The VSM is a mathematical model designed to portray an n -dimensional space, where entities are described by *vectors* with n coordinates in a real space \mathbb{R}^n . A data-centric vector space is a very general and flexible abstraction to model a set of observations, like occurrences of a word in a document. Vector components represent raw or modified observations, which are typically represented as a sparse matrix, where items (e.g., documents) are rows and observations (e.g., word occurrences) are columns. Figure 3 shows an abstract representation of two vectors \vec{v} and \vec{w} in a 2-dimensional space.

Vector spaces inherit a set of particular characteristics from algebraic structures: vectors in the VSM are commutative and associative under addition and can be multiplied by scalars in space. The vectors’ length can be normalized to transform them into *unit vectors* – i.e., vectors of length 1. Coordinates of a unit vector are computed by dividing all vector’s components by the normalization factor, as shown in Equation 5. Figure 3b shows how the vectors from Figure 3a are transformed into unit vectors, while preserving the angle between them.

$$\vec{v} = (v_1, v_2, \dots, v_n) \quad \|\vec{v}\|_2 = \sqrt{\sum_{i=1}^n v_i^2} \quad (5)$$

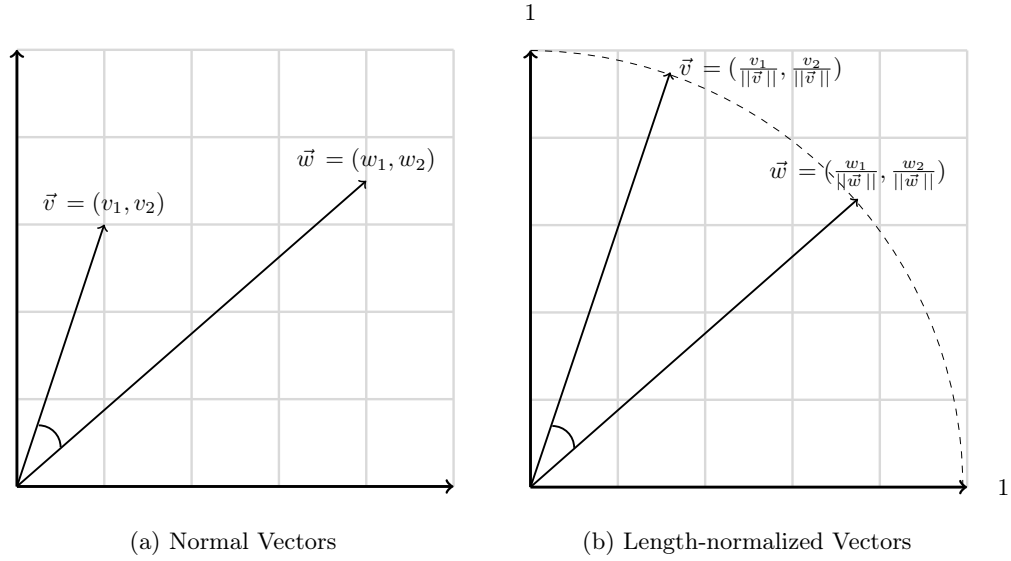


Figure 3: Representation of vectors in the vector space model

3.2.1 The VSM in an Information Retrieval Context

The first use of the vector space model in the context of Information Retrieval (IR) is attributed to Gerard Salton in a 1979 book [Sal89] on automatic text processing.⁵ In the context of information retrieval systems, VSM was introduced to get away from the boolean and probabilistic models. Salton et al.’s IR model [SWY75] uses the vector space model as follows:

1. Extract all words from a machine-readable text;
2. Select a subset of words deemed significant enough to represent the document’s content (meaning);
3. Assign a numeric weight to each word and create a vector of words (i.e., *term-vectors*);
4. Compute a measure of association between the pairs of term-vectors.

As such, in information retrieval, both documents and queries can be represented using ordered term sets (vectors), where the components of vectors are numbers, representing the importance of each term with respect to the document or query, or simply the presence or absence of a term using 1 and 0, respectively. Using such a representation, as Bookstein and Cooper define in [BC76], “*a retrieval operation is a mapping between the space of query words and the space of documents.*” Note that Salton’s model would only work for retrieval systems, where the term-vectors (sets) have a well-defined complement and the request space (query) is identical with the object space (documents).

⁵Early mentions of the VSM model are citing a 1975 article by Gerard Salton, which interestingly seems to not exist in any bibliographical repository. Almost thirty years after Salton’s phantom article, Dubin argues that Salton merely used the VSM as models of specific computations in his earlier papers, rather than a general IR model, until the publication of his 1979 book. Refer to “*The Most Influential Paper Gerard Salton Never Wrote*” [Dub04] for more details.

For example, this model would work for an IR system in which both the query and documents are in the same natural language, like English.

Document Space Configurations

Salton et al. [SWY75] define the document space configurations as follows: Consider a document space consisting of a set of documents $D = (d_1, d_2, \dots, d_n)$, where each document contains one or more terms $T = (t_1, t_2, \dots, t_j)$. In a t -dimensional space, where t is size of the vocabulary, each document d_i is represented by a t -dimensional vector. Each vector component v_i represents the weight of the n th term in the document. Of course, considering all words, numbers and symbols in documents written in a natural language results in a very large number of dimensions for the document space. Certain pre-processing on the documents' content can help to reduce the document space dimensions. Removing *stop words* from documents is one such pre-processing step. Stop words are short function words, such as *the*, *at*, *is*, which appear in almost all documents in the space and bear no discriminating value in a retrieval model. In addition, all word inflections (i.e., modifications of words to express grammatical categories) are normalized to the canonical root of each word during a process known as *lemmatization*. From the set of remaining lemmatized words, a subset of terms are selected, each one is assigned a numeric weight according to its discrimination value, and the term-vectors are constructed. The method used to select designated terms varies in each IR system, based on its underlying design.

Computing a Similarity Coefficient

Given the term-vectors of two documents (or a query and a document), it is possible to compute a similarity coefficient between them. In VSM, the distance between two vectors in a hyperspace reflects the degree of similarity between the entities they represent. The distance of document vectors in space are an inverse function of their similarity. In other words, vectors with smaller distance represent documents with similar content. As opposed to using the Euclidean distance of vectors, which would be large for vectors of different lengths, in VSM we use the cosine of the angle between a pair of normalized term-vectors, since cosine is a monotonically decreasing function for the $[0^\circ, 180^\circ]$ interval and more efficient to compute. Equation 6 shows how the cosine similarity of a document (\vec{d} vector) and a query (\vec{q} vector) is calculated. Note that in IR, we essentially treat the query as a short document.

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{\vec{q}}{|\vec{q}|} \cdot \frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{|v|} q_i \cdot d_i}{\sqrt{\sum_{i=1}^{|v|} q_i^2} \cdot \sqrt{\sum_{i=1}^{|v|} d_i^2}} \quad (6)$$

The q_i and d_i in Equation 6 are the weights of the i th term in the query and document, respectively. The weights are calculated using the terms' frequency in each document and the entire document space, known as the *tf-idf* approach [MRS08]. *Term Frequency* (*tf*) is the number of occurrences of a term in a document. *Inverse Document Frequency* (*idf*) is the inverse of the number of term occurrences across the document space. Naturally, *idf* gives a higher weight to rare terms that contribute a higher discrimination value. Values of *tf* and *idf* are computed using Equations 7 and 8, respectively:

$$\text{tf} = 1 + \log(\text{tf}_{t,d}) \quad (7)$$

$$\text{idf} = \log \frac{N}{\text{df}_t} \quad (8)$$

where $\text{tf}_{t,d}$ is the frequency of term t in document d , df_t is the document frequency of term t in the document space, and N is the total number of documents in the document space. The *tf-idf* value of each term in the vector is then simply calculated as $\text{tf}_{t,d} \times \text{idf}_t$.

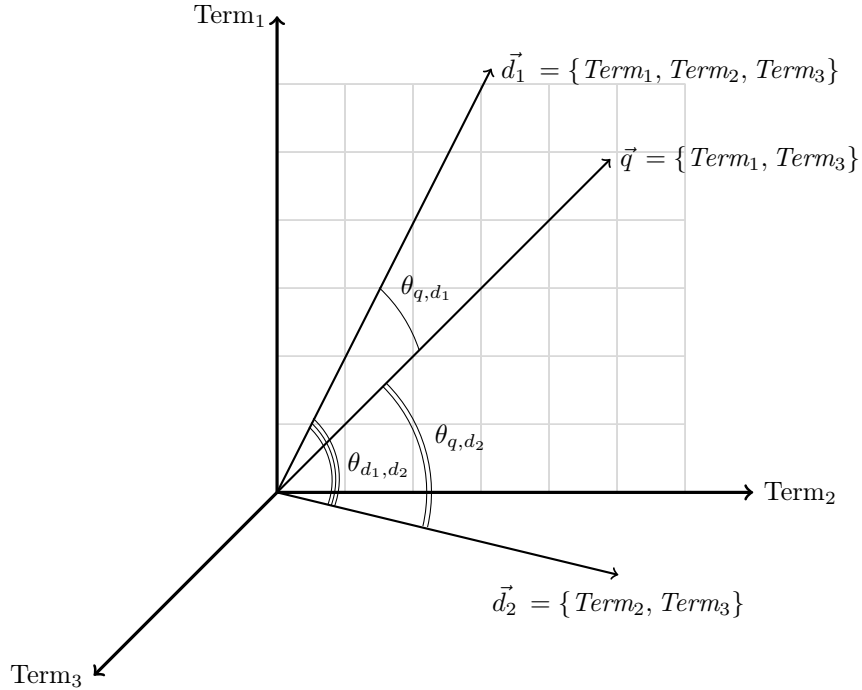


Figure 4: Representation of document vectors in a 3-dimensional space

Figure 4 shows the projection of two document vectors and a query vector into a 3-dimensional

space. All shown vectors are normalized and the cosine of the θ angle between each two vectors is used as a measure of their similarity.

Additionally, many modern search algorithms use a modified version of tf-idf term weighting, referred to as the *Okapi BM25* model [RW94, SJWR00]. While the classical tf-idf model assigns different weights to terms based on their distribution across the document space, it does not distinguish two documents containing the term from one another. BM25, instead, takes on a probabilistic approach towards IR and computes the term frequency based on the length of the document containing it, with respect to the average length of documents in the entire document space. Thus, “while a longer document has more words, each individual word has the same probability of being the term in question” [RW94].

3.2.2 Evaluation and Metrics

The effectiveness of information retrieval systems is determined based on the notion of *relevance*. Relevance is a binary assessment of whether a retrieved document addresses a user’s information need, which is generally expressed in form of a query [MRS08]. Similar to NLP systems evaluation, given an expression of the information need, the assessment is performed against a *goldstandard* (or *ground truth*) that contains the relevant documents for each corresponding query. In practice, IR systems are trained for maximized performance on a subset of the goldstandard, commonly referred to as the *development set* and then run on a *test set* for obtaining a generalized, unbiased estimate of their performance on previously unseen data. The standard practice is to compare the relative effectiveness of different information retrieval methods against the same test collection.

The quality assessment of IR systems can be regarded from two perspectives: (i) the relevance of the documents retrieved, and (ii) the order in which they are presented to a user. The first perspective is concerned with finding all and only relevant documents from the given document space, whereas the latter perspective penalizes the system’s performance for presenting the user with relevant documents, but in lower ranks (positions) in a result set.

Evaluation of Unranked Results

Based on the relevance binary scale, two basic measures are reported in IR evaluations, namely *Precision* and *Recall*. Precision is the fraction of retrieved documents that are relevant, whereas recall is the fraction of relevant documents that are retrieved. Precision and recall are calculated based on the contingency matrix showed in Table 3, using the formulas in Equations 1 and 2.

Table 3: Information retrieval systems evaluation contingency table

Predicted Actual	Relevant	Irrelevant
	Retrieved	Not Retrieved
	True Positive (TP)	False Positive (FP)
	False Negative (FN)	True Negative (TN)

TP : Correctly identified documents

TN : Correctly rejected documents

FP : Incorrectly identified documents (i.e., false alarm)

FN : Incorrectly rejected documents (i.e., misses)

Evaluation of Ranked Results

The evaluation of information retrieval systems, where ordering of the result set is important, is based on the premise that relevant documents appearing lower in an ordered result set must be penalized. The Discounted Cumulative Gain (DCG) [MRS08] is a metric borrowed from machine learning approaches applied to ranking that reports a single measure for a system's results at a cut-off rank, particularly when the notion of relevance is non-binary. The DCG for a ranked result set at position p is calculated using Equation 9:

$$DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2 i} \quad (9)$$

where rel_i is the relevance of a document retrieved at position i in the result set and the denominator of the equation is a logarithmic reduction factor to penalize lower ranks.

When comparing a system across different queries, it is a common practice to normalize the DCG (since not all queries retrieve the same number of relevant documents), calculated using Equation 10:

$$nDCG_p = Z_p \sum_{p=1}^N \frac{2^{r(p)} - 1}{\log_2(1 + p)} \quad , \quad r(p) = \begin{cases} 1 & \text{if document at position } p \text{ is relevant} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where Z_p is a normalization constant calculated such that an ideal ordered results set would obtain an nDCG of 1; and $r(p)$ is a function of relevance for a document retrieved at position p .

3.3 Linked Data Principles

The vision and architecture of the web of *linked data*, in essence, is closely related to the principles of the World Wide Web. There exists a myriad of data available on the web in isolated fragments. The rationale behind linked data is to foster the *reuse* of this data, through inter-connecting resources with typed (semantic) links. The Resource Description Framework (RDF) – a W3C recommendation⁶ – provides a standard mechanism for [HB11] “*specifying the existence and meaning of connections between items described in this data.*” Unlike the World Wide Web, the web of linked data consists of *things* and not just (web-based) documents. RDF provides a way to describe things, like persons or abstract concepts, and how the defined entities are (semantically) related to each other, e.g., that a person is the author of an article. RDF Schema (RDFS)⁷ provides a *semantic* extension for RDF’s basic vocabulary to describe group of things – referred to as *classes* – and the relationship between resources using a *property* system. Adhering to the following linked data principles [BL06] facilitates reuse, share and discovery of relevant data within disparate datasets:

- Using Uniform Resource Identifier (URIs)⁸ as names for things.
- Using the Hyper Text Transfer Protocol (HTTP) to publish the data, so clients can look up the URIs.
- Providing useful, relevant information when a client looks up a URI.
- Including links to other relevant data, so clients can discover more things.

Such a hyperlinked structure creates a decentralized data space that can be navigated through incoming and outgoing links by both humans and machines alike. When a URI is *dereferenced* over HTTP, a description of the identified object is returned to the client after a *content negotiation* step. The content negotiation mechanism allows clients to express what kind of response they prefer. For example, a human user prefers to read a user-friendly HTML webpage with hyperlinks that she can click on, while an agent (machine) would prefer the RDF description of the dereferenced entity with incoming and outgoing semantic links.

3.3.1 Web of Linked Open Data

The Linked Open Data⁹ project is the result of collective efforts of the Semantic Web community since 2007 to identify existing data on the web and publish them in a linked data compliant format. Essentially, Linked Open Data (LOD) is [BL06] “*linked data which is released under an open license,*

⁶Resource Description Framework, <https://www.w3.org/RDF/>

⁷RDF Schema 1.1, <https://www.w3.org/TR/rdf-schema/>

⁸In the case of URIs that contain characters from the Universal Character Set, like Unicode, the W3C suggests to use the Internationalized Resource Identifiers (IRIs) instead, which can encode characters beyond the ASCII set.

⁹Linked Data, <http://linkeddata.org>

which does not impede its reuse for free.” The Semantic Web community efforts culminated in a massive web of inter-connected datasets, including governmental, social networks, geographical and life sciences data, known as the Linked Open Data Cloud.¹⁰ The LOD cloud quickly grows as more open datasets emerge on the world wide web. In fact, the size of the LOD cloud has grown 90-fold in the last decade,¹¹ providing access to around 10,000 datasets with 192 million triples in multiple languages. Figure 5 shows the current topology of the LOD cloud in the first half of 2017, illustrated as a directed graph, where nodes are open datasets and edges represent incoming and outgoing links between the data instances. The datasets are colour-coded based on their topic (e.g., geography, government, life sciences) and their size is representative of their relative size in total number of triples.¹² The current state of the LOD cloud shows multiple high-density subgraphs, especially in life sciences, with Medical Subject Headings¹³ as the pivotal dataset with the highest connectivity (linkage) to relevant data.

At the heart of the general knowledge subgraph of the LOD cloud shown in Figure 5 is the DBpedia Ontology.¹⁴ Serving as the “*nucleus for a web of open data*” [ABK⁺07], it comprises the knowledge contained in the Wikipedia¹⁵ articles in a structured format. Although primarily developed for machine consumption of its information, it also serves a human-readable version of its knowledge base for web-based browsing. Figure 6 shows the DBpedia entry page for the <dbpedia:Text_mining> topic.¹⁶ The current version of the DBpedia Ontology provides rich RDF descriptions of 6.6 million entities of geographical, persons, companies, books, and scientific publication types in multiple languages. Where applicable, DBpedia entities are linked to relevant data in 50 external knowledge bases, such as YAGO [SKW07], DBLP,¹⁷ UMBEL¹⁸, among others, allowing humans and machines to start from an entity in the DBpedia ontology and traverse the web of linked open data to other relevant data sets.

3.3.2 Evaluation

Linked Open datasets are evaluated based on a qualitative scale, rather than, say, their size. Tim Berners-Lee defined a 5-star rating scale [BL06] for assessing the quality of linked open datasets:

★ Data is available on the Web under an open license

★★ Data is available in a machine-readable, structured format

¹⁰Linked Open Data Cloud, <http://lod-cloud.net>

¹¹Linked Open Data Statistics, <http://stats.lod2.eu>

¹²A high-resolution, interactive map of the LOD cloud is available at <http://lod-cloud.net/versions/2017-02-20/lod.svg>.

¹³MeSH Ontology, <https://www.nlm.nih.gov/mesh/>

¹⁴DBpedia Ontology, <http://wiki.dbpedia.org/services-resources/ontology>

¹⁵Wikipedia, <https://www.wikipedia.org>

¹⁶‘Text Mining’ in the DBpedia Ontology, http://dbpedia.org/resource/Text_mining

¹⁷DBLP, <http://dblp.dagstuhl.de/>

¹⁸UMBEL, <http://umbel.org/>

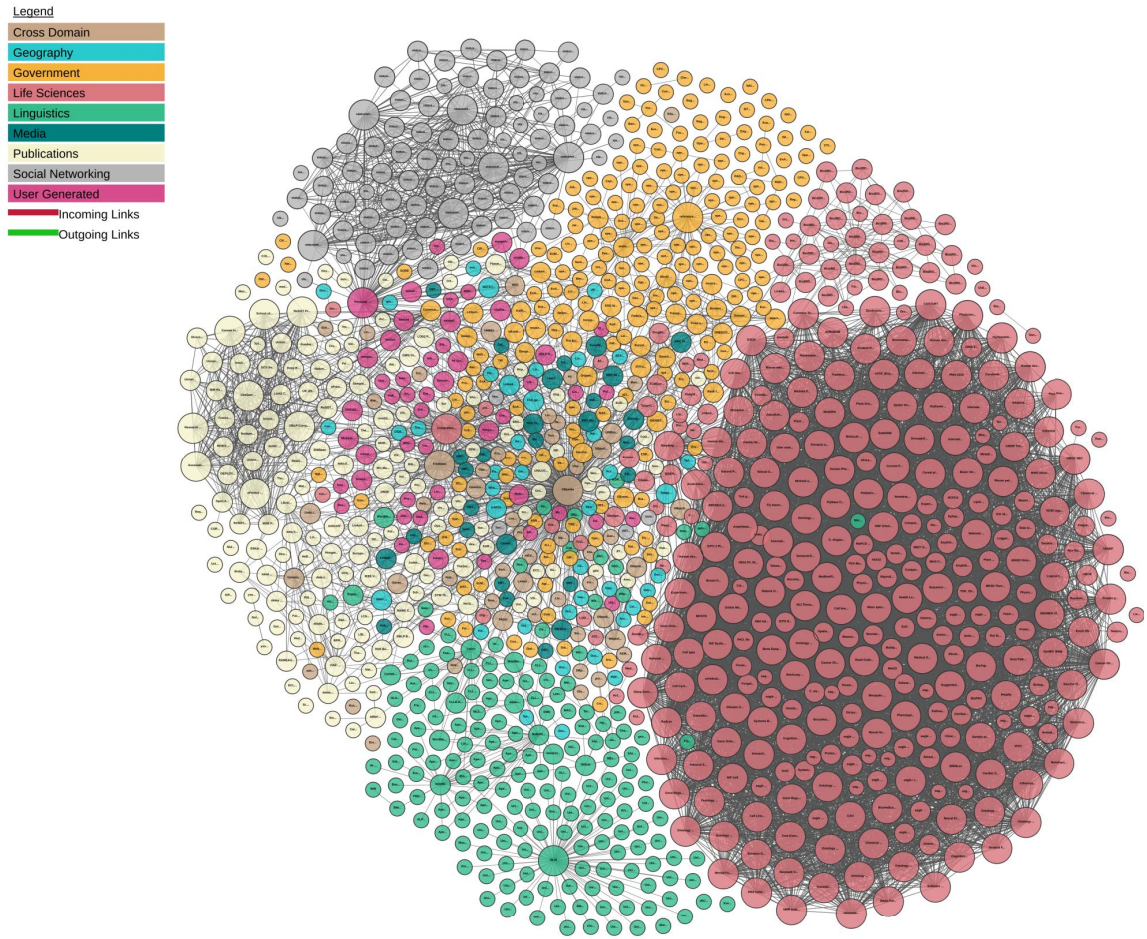


Figure 5: Topology of the web of Linked Open Data (LOD) in 2017 [AMB⁺17]

- ★★★ Data is available in a non-proprietary format
- ★★★★ Data is described using standards from the W3C (e.g., RDF)
- ★★★★★ Data is linked to other people's data to provide context

3.4 User Modeling and Recommender Systems

Recommender systems aim at predicting users' behaviour or providing custom-tailored content for users through collecting their preferences and interactions with the system. The two most common approaches used in recommender systems are *content-based* and *collaborative* filtering. In the former approach, items are recommended based on their similarity to a user's characteristics, whereas in a collaborative filtering approach, the system aims at clustering like-minded users and predicting a






 Browse using  Formats  Faceted Browser  Sparql Endpoint 	
<h2>About: Text mining</h2> <p>An Entity of Type : CodingSystem106353757, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org</p>	
Property	Value
dbp:abstract	<ul style="list-style-type: none"> ■ Text mining, also referred to as text data mining, roughly equivalent to , refers to the process of deriving high-quality information from text. [...] (en)
dct:subject	<ul style="list-style-type: none"> ■ dbc:Data_mining ■ dbc:Artificial_intelligence_applications ■ dbc:Computational_linguistics ■ dbc:Natural_language_processing ■ dbc:Statistical_natural_language_processing ■ dbc:Data_analysis
rdf:type	<ul style="list-style-type: none"> ■ owl:Thing ■ yago:Application106570110 ■ yago:CodingSystem106353757 ■ yago:Software106566077 ■ yago:WikicatArtificialIntelligenceApplications
rdfs:label	<ul style="list-style-type: none"> ■ Text mining (en)
owl:sameAs	<ul style="list-style-type: none"> ■ wikidata:Text mining ■ dbpedia-nt:Text mining ■ dbpedia-wikidata:Text mining ■ yago-res:Text mining

Figure 6: Human-readable version of a DBpedia entity

user’s behaviour based on her similarity to other users. In this dissertation, we solely focus on the content-based recommendation techniques relevant to our agent’s services.

In a content-based filtering approach, each item is represented as an abstract set of features, which are the terms extracted from an item’s description. The same approach is employed in modeling users, where the terms referring to a user’s interactions with the system (e.g., likes, dislikes) or extracted from the description of items, for instance, in their purchase history, are stored in a so-called *user profile*. Essentially, a user profile is an instance of a user model that contains either a user’s characteristics, such as knowledge about a topic, interests and background, or focuses on the context of a user’s work, e.g., location and time [BM07]. Depending on the application offering personalized content, different features are taken into account during a modeling process. Brusilovsky and Millán [BM07] enumerate the five most popular user profiling features found in literature, when viewing the user as an individual, namely:

Knowledge captures an individual’s understanding of a domain subject, typically on a quantitative (from 0 to 5) or qualitative (from ‘poor’ to ‘good’) scale.

Interests constitute a set of domain subjects that an individual wants to learn about or obtain

related information on, with a *weight* feature attached to each topic representing their significance to the user. The granularity of the topics of interest is determined by the adaptive system use cases.

Goals and Tasks express the “*immediate purpose for a user’s work*” [BM07] within the adaptive system. Examples of goals or tasks include an information need (in a search application) or a learning goal (in an educational system).

Background contains a set of domain subjects related to the individual’s past experiences. The background features can be derived from the user’s profession (e.g., professor vs. student), responsibilities (e.g., writing a manuscript vs. finding related works), past work (e.g., academic publishing record) or language ability (e.g., native vs. non-native speaker).

Individual Traits is an aggregation of personal characteristics of an individual that can distinguish the user from others within the same stereotype. Examples of traits are cognitive styles (e.g., user’s habit in organizing information), learning modality (e.g., visual vs. content-based) or accessibility (e.g., user’s device of choice or physical impairments) concerns.

Once a user profile is populated with relevant terms, a recommender system then uses the two sets of features (for users and items) in order to measure an item-to-item or item-to-user similarity metric and use them to decide whether a recommendation must be made to the user.

3.5 Agents and the Semantic Web

Before we explain the design of personal research agents in this dissertation, we must clarify the definition of an agent. The notion of autonomous agents is as old as the artificial intelligence field and a host of definitions exist in the relevant literature (see [FG97] for a review). The definition by Pattie Maes, a pioneer in this area, describes autonomous agents as [Mae95] “*computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed.*” Franklin and Graesser [FG97] propose a formal definition for an autonomous agent and what clearly distinguishes a software agent from just any program: “*An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.*” Adhering to this definition, for a software application to constitute an (autonomous) agent, it must perceive its *environment*, be *reactive* in a timely fashion to respond to changes that occur, and *without* the direct intervention of humans. Therefore, under the Franklin and Graesser taxonomy, all software agents are programs by definition, but not all programs are agents. It should also be noted that, as perceived by some, agents are ‘typically much

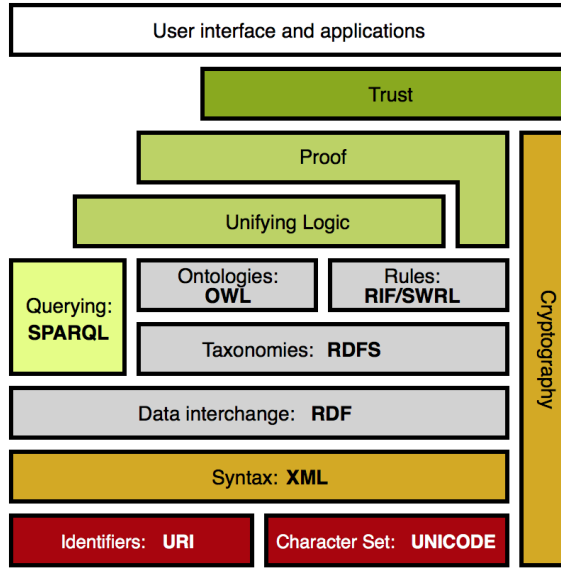


Figure 7: The semantic web stack (proposed by Tim Berners-Lee)

smaller’ than multifunction (software) applications, as their design is dedicated to accomplish a set of pre-defined tasks.

The multi-layer architecture of the Semantic Web¹⁹ shown in Figure 7 provides a common ground for agents to discover, aggregate and exchange information with adequate expressivity using available ontological resources. Consequently, agents working on the semantic web increasingly rely on a shared understanding of resources, tasks and their environment so that they can find possible ways of fulfilling their users’ needs – a vision on par with the original outlook of the Web’s early days [Hen01]. Deriving from this idea, personal research agents are autonomous software agents that can support scholarly users, like researchers, in their tasks through semantic analysis and synthesis of diverse scientific resources, such as articles or datasets. Thereby, such agents can proactively exploit large-scale, high-performance computing techniques to alleviate the users’ information overload, caused by the rapid growth of scientific dissemination, and devise new approaches for scientific discovery.

With this definition in mind, the purpose of this dissertation is not to introduce a new Multi-Agent System architecture; Rather, we aspire to envision, design and realize agents that can integrate the knowledge contained within scholarly literature with available ontologies on the web and help users in finding, reading and understanding tasks in a typical research workflow. Nevertheless, the presented research can set the stepping stones of the grand vision for a web of autonomous science bots.²⁰

¹⁹The Semantic Web Stack, <https://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>

²⁰Refer to [Kuh15] for a position paper on a model for the future of scientific computation.

3.6 Summary

The work presented in this dissertation is a novel fusion of several techniques from disparate domains. In this chapter, we provided a cursory introduction to the methodologies used in our approach and detailed how we will evaluate them against a set of objective metrics.

Chapter 4

Literature Review

In this chapter, we provide an overview of existing efforts related to our research. We introduce the relatively new domain of semantic publishing and review relevant works on automatic capturing of scientific literature’s argumentative structures. We further look into how such constructs can be enriched with semantic vocabularies and used in recommender systems. Note that the analysis of inter-document bibliographical metadata, such as citation networks, is out of the scope of the presented research and thus excluded from our review.

4.1 Semantic Publishing

Semantic publishing is the practice of publishing information on the Web using semantic web technologies, such that computers can read the structure and meaning of its content. Shotton [Sho09] expands the semantic publishing definition to augmenting documents with *“anything that enhances the meaning of a published journal article that facilitates its automated discovery, enables its linking to semantically related articles, provides access to data within the article in actionable form, or facilitates integration of data between papers”*. In another article titled *The Five Stars of Online Journal Articles* [Sho12], Shotton characterizes *“the potential for improvement to the primary medium of scholarly communication”* using semantic web technologies. He reviews the status quo of the online dissemination of scientific literature and formulates five factors for improving online journal articles using web technologies, including enrichment of content and publishing them as machine-readable data. Shotton is a member of the advisory committee of FORCE11,¹ a community of scholars, librarians, archivists and publishers with the mission of *“facilitating change towards improved knowledge creation and sharing”* [BCD⁺12]. FORCE11 was formed after a series of key meetings in 2011 that contributed to the formulation of a number of state-of-the-art prototypical systems and scientific

¹FORCE11, <https://www.force11.org>

venues for semantic publishing.

*Beyond the PDF*² is a series of workshops by the FORCE11 community with a focus on the future of research communication and e-Scholarship. The goal of the Beyond the PDF 1 (2011) & 2 (2013) workshops was to develop a mandate, as well as open source tools, to facilitate knowledge sharing and discovery among researchers. The workshop proceedings cover a variety of topics in semantic publishing by researchers and practitioners of the publishing industry, like Elsevier.³ These topics include issues of current publishing paradigms, extracting discourse elements from scientific literature, next generation of document production and reading tools, among others.

*SePublica*⁴ was a series of workshops collocated with the Extended Semantic Web Conference (ESWC),⁵ dedicated to semantic publishing. SePublica started in 2011, with the mission of gathering researchers and practitioners of semantic web technologies to discuss and propose ideas on transforming the web from a dissemination platform to a web of interconnected documents with machine-readable information. Each year, the SePublica workshop had a focused theme, like exploring the future of scholarly communication (in 2012) and bridging the gap between publications and data (in 2013). ESWC launched its first *Semantic Publishing Challenge* in 2014, for producing and exploiting semantic metadata available in linked open datasets about scientific publications (like DBLP⁶), which is held on an annual basis (see Section 8.1.1). Starting in 2015, the SAVE-SD workshop series⁷ replaced other semantic publishing venues, with a focus on enhancing scholarly data.

The World Wide Web Consortium (W3C) created a *Scientific Discourse Task Force*⁸ in 2011 with the mission of providing a semantic web platform for linking discourse elements in biomedical literature to biological categories specified in LOD ontologies. The task force is composed of semantic publishing researchers and biomedical experts that develop and align ontologies to formalize discourse elements, rhetorical structures and experiments in digital scientific communication. The ultimate goal of this task force is to develop an architecture for automatic mining of treatment outcomes from literature as linked data for meta-studies of drug efficiency on neurodegenerative disorders, like Alzheimer.

4.1.1 Scientific Literature Markup

An essential foundation of the semantic publishing process is the existence of vocabularies that are shared by a target community with an unambiguous, formalized meaning for machines. Controlled

²Beyond the PDF, <https://sites.google.com/site/beyondthepdf/>

³Elsevier, <https://www.elsevier.com>

⁴SePublica Workshop, <http://sepublica.mywikipaper.org/>

⁵Extended Semantic Web Conference, <http://eswc-conferences.org>

⁶DBLP, <http://dblp.kbs.uni-hannover.de/dblp/>

⁷Semantics, Analytics, Visualisation: Enhancing Scholarly Data (SAVE-SD), <http://cs.unibo.it/save-sd/>

⁸W3C Scientific Discourse Task Force, <http://www.w3.org/wiki/HCLSIG/SWANSIOC>

vocabularies mandate the use of pre-defined lists of words or phrases to tag units of information. In scientific literature mining, special markup is added to text (either manually or automatically) to describe its content using available vocabularies on the semantic web.

Semantic markup refers to the additional information attached to any content in order to represent its ‘nature’ and relationships between its units.⁹ Early efforts of semantic markup in scientific literature are related to adding explicit markup of mathematical information to manuscripts. MathML¹⁰ (now part of HTML5) and OpenMath [CC99] are two examples of semantic markup languages used to separate the presentation and meaning of mathematical formulas in web pages and scientific manuscripts. STeX [Koh08] is a collection of L^AT_EX macro packages for pre-loading documents with a formalization of mathematical content into a Mathematical Knowledge Model (MKM) compliant format, so that machines can operate on them. An underlying formalism (i.e., ontology) defines concepts of different granularity, from object-level annotations, like complex numbers, up to theory-level annotations representing sets of axioms. These markup languages, however, have very limited applicability and have not found adoption outside of the mathematics domain literature. In the following sections, we focus our survey on markup languages that can be applied to scientific literature, irrespective of their concrete domain.

Structural Markup

Prior to the analysis of scientific literature for their latent knowledge, we first need to provide the foundation for a common representation of documents, so that (i) the variations of their formats (e.g., HTML, PDF, L^AT_EX) and publisher-specific markup can be converted to one unified structure; and (ii) various segments of a document required for further processing are explicitly annotated, e.g., by marking up tables’ content separately from the document’s main matter.

The eXtensible Markup Language (XML) is used by many semantic markup projects. One notable work is the University of Cambridge’s SciBorg project [CCMR⁺06], which proposes a common XML markup, called SciXML [RCTW06], for domain-independent research papers. It contains a set of vocabularies that separate a document into sections and paragraphs that may contain references, footnotes, theorems and floats, like tables and figures. SciXML also provides a stand-off¹¹ annotation formalization to represent various linguistic metadata of a given document, for example, for encoding chemical terms.

Other existing vocabularies are not specific to the structure of scientific literature, but provide the vocabulary for associating metadata to documents. The Annotation Ontology (AO) [COGC⁺11]

⁹Web Content Accessibility Guidelines (WCAG), <http://www.w3.org/TR/WCAG20-TECHS/>

¹⁰MathML, <http://www.w3.org/Math/>

¹¹In stand-off annotation style, the original text and its annotations are separated into two different parts (files) and connected using text offsets.

enables a stand-off annotation schema to annotate scientific documents on the Web. It defines classes, like annotations and annotation sets, as well as text selectors to mark up string positions in a document and uses XPointer¹² to attach the annotations to specific parts of an XML document. It also features a provenance model¹³ through its integration with the FOAF¹⁴ vocabulary and a set model for specifying groups and containers of annotations.

The W3C Open Annotation Community¹⁵ is a working group aiming towards a common specification of an annotation schema for digital resources in RDF format. Its work is mainly focused on a reconciliation of the Annotation Ontology (AO) and the Open Annotation Model¹⁶ (OAM) developed based on the W3C Annotea¹⁷ project (which has a more restricted set of metadata for sharing web documents). In contrast to other annotation schemas, the focus of the OAM is on sharing annotations for scholarly purposes with a baseline model of only three classes: a *Target* being annotated, a *Body* of information about the target, and an *Annotation* class that describes the relationship between the body and target, all with dereferenceable URIs.

Although the Open Annotation Community is trying to align the existing annotation ontologies for their use in scholarly publications, the document vocabularies have been developed in disjointed efforts and were motivated by project-specific needs, like SciXML for literature in the chemistry domain. Moreover, annotation vocabularies treat the documents as unrelated fragments of text, whereas in scientific literature this is obviously not the case – the sections of an article follow a logical, argumentative order. Peroni [Per12] has a similar observation and makes a distinction between XML-like languages for *document markup* and *semantic markup* like RDF (which may use XML as a serialization format). He argues that document markup languages leave the semantics of the content to human interpretation (and in the worst case, of the markup itself) and lack “*expressiveness for the multiple and overlapping markup on the same text*”. Peroni et al. introduced the EARMARK [DIPV09] markup meta-language that models documents as collections of addressable text fragments and associates their content with OWL assertions to describe their structural and semantic properties. Peroni is also an author of the DoCO¹⁸ ontology, which is part of the SPAR (Semantic Publishing and Referencing) ontology family [SPKM09]. The DoCO ontology specifically defines components of bibliographic documents, like the main matter of books and theses, chapters, figures, and bibliography sections, enabling their description in RDF format.

Other relevant vocabularies are Dublin Core and the Bibliographic Ontology (BIBO), referenced

¹²XPointer, http://www.w3schools.com/xml/xml_xpointer.asp

¹³Provenance metadata describe entities and processes involved in producing a piece of information, e.g., the author of an annotation, and are used to analyze its origins and authenticity.

¹⁴The FOAF Project, <http://www.foaf-project.org>

¹⁵The Open Annotation Community, <http://www.w3.org/community/openannotation/>

¹⁶Open Annotation Model, <http://www.openannotation.org/spec/core/>

¹⁷W3C Annotea Project, <http://www.w3.org/2001/Annotea/>

¹⁸The Document Components Ontology (DoCO), <http://purl.org/spar/doco>

in this dissertation. Dublin Core¹⁹ is the oldest schema with the widest coverage of document metadata, as it was designed as a broad and generic vocabulary for describing a wide range of resources, not just textual documents. BIBO²⁰ is a shallow ontology that provides a vocabulary for describing citations and bibliographic references in RDF. Compared to the two aforementioned ontologies, DoCO is relatively new, but was designed ground up for semantic publishing workflows. Moreover, DoCO is an OWL 2 DL vocabulary, covering both a description of document layers and discourse. It is “*a suite of orthogonal and complementary ontologies*” for describing different aspects of the semantic publishing domain, providing vocabularies for resources, roles and workflows in the domain.

Rhetorical Entities Markup

In this section, we describe other existing vocabularies used for annotation of rhetorical entities in scientific literature. In addition to the document structure vocabulary, the DoCO ontology described in the previous section also provides a vocabulary of rhetorical blocks in scientific literature, like Background, Conclusion and Discussion, through reusing the SALT Rhetorical Ontology and Discourse Elements Ontology.²¹

CoreSC [LTSB10] takes on a different approach of annotating scientific documents. It treats scientific literature as a human readable representation of scientific investigations and therefore, has a vocabulary that pertains to the structure of an investigation, like an Experiment or Observation. CoreSC is itself a subpart of the EXPO [SCSK06] ontology, a comprehensive vocabulary for defining scientific experiments, like Proposition or Substrate. While ontologies like SALT or AZ-II [TSB09] focus on the rhetorical structure of a document, ontologies like CoreSC and EXPO are used for supporting reproducibility in domains, like Chemistry or the *omics* sciences.

The Ontology of Rhetorical Blocks (ORB)²² is another W3C work-in-progress aiming at a formalization of the rhetorical structure of scientific publications. ORB provides a set of classes for a coarse-grained identification of rhetorical zones in a document, such as Headers, Introduction, Methods and Results. The ORB ontology provides a set of guidelines²³ for the further decomposition of the ontology to finer-grained elements carrying a rhetorical role in a text, e.g., the SWAN²⁴ and SIOC²⁵ ontologies developed by the W3C Scientific Discourse Task Force described in Section 4.1.

¹⁹Dublin Core, <http://dublincore.org>

²⁰Bibliographic Ontology (BIBO), <http://bibliontology.com>

²¹Discourse Elements Ontology, <http://purl.org/spar/deo>

²²Ontology of Rhetorical Blocks (ORB), <http://www.w3.org/2001/sw/hcls/notes/orb/>

²³ORB Extension, <http://www.w3.org/2001/sw/hcls/notes/orb/#extensions>

²⁴The SWAN Ontology, <http://www.w3.org/2001/sw/hcls/notes/swan/>

²⁵Semantically-Interlinked Online Communities, <http://www.w3.org/2001/sw/hcls/notes/sioc/>

4.1.2 Discussion

Our overview of the semantic publishing research community shows that the realization of machine-understandable documents is made possible through a synergy of efforts from multiple areas: The semantic web community is defining and aligning vocabularies and ontologies on the web with the help of domain experts, such as biologists, to provide for a shared, unambiguous understanding of domains concepts and their attributes. The NLP developers work hand in hand with domain experts to implement automatic solutions for detecting concepts in scientific literature and linking them to their corresponding resources on the web of data. These efforts include the development of vocabularies to mark up the structure of documents and text mining tools that annotate rhetorical entities, like *Claims* or *Speculations*, in scientific literature. However, the existing scholarly-specific vocabularies are either very restrictive (e.g., EXPO only provides vocabularies for scientific experiments) or they are in their early stage (e.g., DoCO is less than 3 years old) and have not yet been widely adopted in any automated knowledge discovery tools. Subsequently, we need to enhance existing vocabularies or develop our own for integration in text mining solutions that use modern linked data vocabularies in the extraction phase. When the identified entities are explicitly assigned with their semantic types, a knowledge base of scientific literature can be created that combines the vocabularies of documents, rhetorical entities and their annotations, to serve a variety of tasks, like creating a summary of documents based on their contributions, for a user.

4.2 Argumentation Mining in Scientific Literature

Argumentation mining aims at automatically detecting and classifying argumentative propositions in text [MM11]. Primarily popular in the analysis of legal corpora, argumentation mining techniques have found their way into scientific publishing, considering its unprecedented growth rate.

4.2.1 Proposed Schemas

Prior to creating automatic approaches for argumentation mining, a schema for the type and structure of argumentative propositions in scientific literature had to be established first. While there seems to be no consensus on a definitive schema, a number of works have proposed various schemas, some of which became quite popular among researchers. An overview of the schemas discussed in this section is provided in Table 4.

Swales’ “*Create A Research Space*” (C.A.R.S) model [Swa90] is one of the earliest works that proposed a set of discipline-based writing practices for research articles, later adopted as a schema for the classification of propositions in an article into one of its three so-called rhetorical *moves*, each with their own *steps*. On a high level, Swales suggested that researchers have to make at least

Table 4: Overview of existing schemas for rhetorical blocks in scientific literature

Related Work	Schema Categories
Swales [Swa90]	ESTABLISHING A TERRITORY, ESTABLISHING A NICHE, OCCUPYING THE NICHE
Liddy [Lid91]	PURPOSE, METHODOLOGY, RESULTS, CONCLUSIONS, REFERENCES
Kando [Kan97]	PROBLEMS, EVIDENCE, ANSWERS
Anthony [Ant99]	Swales model plus ' <i>Step 3-3: Evaluation of research</i> '.
Teufel et al. [TCM99]	AIM, CONTRAST, BASIS, TEXTUAL, BACKGROUND, OTHER, OWN
Groza et al. [GHMD07]	ABSTRACT, MOTIVATION, SCENARIO, CONTRIBUTION, DISCUSSION, EVALUATION, BACKGROUND, CONCLUSION, ENTITIES
de Ribaupierre and Falquet [dRF17]	METHODOLOGY, HYPOTHESIS, RELATEDWORK, FINDING, DEFINITION
de Waard and Tel [dWT06]	ANNOTATIONS, BACKGROUND, CONTRIBUTION, DISCUSSION, ENTITIES
Pendar and Cotos [PC08]	Swales' model plus ' <i>Step1E: Adding to what is known</i> ', ' <i>Step 1F: Presenting justification</i> ', ' <i>Step 2B: Presenting hypotheses</i> ', ' <i>Step 4: Summarizing methods</i> ', and ' <i>Step 6: Stating the value of the present research</i> '
Ruch et al. [RBC ⁺ 07]	PURPOSE, METHODS, RESULTS AND CONCLUSION

three moves in creating a research space for their articles: First, authors have to demonstrate that the general idea of their research is important and interesting. Next, they have to make a clear argument that a particular gap in previous research can be fulfilled by their research or an extension of it. As a final move, authors must announce the means by which their research will contribute new knowledge or extend the previous state-of-the-art. Based on roughly the same schema, Liddy [Lid91], Kando [Kan97] and Teufel [TCM99] also proposed their own argumentation schemas that categorize various authors' rhetorics in scientific literature. Anthony [Ant99] argues from the perspective of a corpus linguist, analyzing the characteristic features of Computer Science articles with regard to how Swales' model is applied on articles that received 'Best Paper' awards in the field of software engineering. Subsequently, Anthony proposed a variation of Swales' C.A.R.S model [Ant99] by adding a new step for authors to describe how they evaluated their research, and modified the original model, such that multiple steps can be combined within one move.

4.2.2 Manual Approaches for Rhetorical Entity Markup

Groza et al. introduced a framework for the semantic annotation of scientific literature. The SALT framework [GHMD07] employs a user-driven approach, where authors manually mark up chunks of text with semantic annotations while they are writing a manuscript. The underlying architecture of SALT is composed of two layers: a *syntactic* layer and a *semantic* layer. The semantic layer defines three ontologies to capture the structural information of a document, as well as the semantic entities

mentioned in its content: a *Document Ontology*²⁶ that defines entities like text blocks, Abstract and Title; a *Rhetorical Ontology*²⁷ that defines concepts like Claims, Explanations and Results; and an *Annotation Ontology*²⁸ that provides the means to attach syntactic and semantic markup to the document. The syntactic layer extends the L^AT_EX writing environment by introducing special commands to mark up both the document structure and rhetorical elements in a source file, according to the three aforementioned ontologies. In the early versions of the SALT framework, the embedded semantic markup was extracted from manuscripts in the compilation phase and visualized in HTML pages generated from the document metadata.

The SALT framework was later extended and adapted for extracting Claims from text with the ultimate goal of creating a knowledge network from scientific publications. Groza et al. introduced ClaiSE [GHMD07] and its successor, the KonneX^{SALT} [GHMD08] system, which provide support for (manual) identification, referencing and querying of claims in a collection of documents. They extended their Rhetorical Ontology with concepts, such as the generalizations of claims and their related text chunks, to provide for identifying claims with multiple possible representations across a dataset. They also introduced a BibTeX-like referencing system for the citation of claims that can be incorporated into the L^AT_EX environment using special commands and queried using a web interface.

Groza et al.’s approach is quite similar to an earlier work by de Waard and Tel [dWT06], published around the same time, which also proposes a L^AT_EX-based approach with which authors can semantically annotate their manuscripts. The authors proposed ‘ABCDE’ as a rhetorical structure for scientific publications that essentially divides a document into three parts, namely (i) Annotations (A) that attach shallow metadata to a document; (ii) Background, Contribution and Discussion (BCD) that mark up the ‘core sentences’ describing the authors’ work; and (iii) Entities (E), like names of persons, projects or references, that can be mined and described using RDF.

4.2.3 Automatic Approaches for Rhetorical Entity Detection

Subsequent works aimed at adopting Mann’s Rhetorical Structure Theory (RST) [WT88] to automatically find argumentations in text and classify them into existing schemas. RST is a descriptive framework for the organization of natural language content by characterizing fragments of text and the relations that hold between them. Automatic rhetorical analysis of text was accelerated by works like Marcu’s rhetorical parser [Mar99] that derives discourse structures from unrestricted text. Subsequent researchers then started looking into rule-based or statistical approaches to detect the rhetorical propositions in scientific literature for different use cases.

²⁶SALT Document Ontology, <http://lov.okfn.org/dataset/lov/vocabs/sdo>

²⁷SALT Rhetorical Ontology, <http://lov.okfn.org/dataset/lov/vocabs/sro>

²⁸SALT Annotation Ontology, <http://lov.okfn.org/dataset/lov/vocabs/sao>

Teufel [Teu99, Teu10] proposed a schema of seven rhetorical types [TCM99] for propositions in scientific documents and identified so-called Argumentative Zones (AZ) from text as a group of sentences with the same rhetorical role. A two-pass Hidden Naïve Bayes classifier trained on 16 positional and syntactical features of sentences in a 80-document corpus of computational linguistics journal articles achieved a $\kappa = 0.48$ Kappa agreement with human annotators (and a macro F_1 -measure of 0.54 with a 7-fold validation) on around 12,500 sentences in her approach. Applications of AZ include automatic summarization [TM02], citation indexing [Teu06] and machine-generated feedback on writing tasks [FPT⁺04].

Anthony and Lashkia [AL03] introduced *Mover*, a machine learning tool with a graphical interface that can analyze a given text and classify it into one of the steps of their proposed schema (see Table 4). *Mover* was evaluated on a corpus of 100 Computer Science journal abstracts using a Naïve Bayes classifier trained on a ‘bag of clusters’, i.e., uni- to penta-grams of words in a document. A 5-fold validation on his testing dataset resulted in an average 68% accuracy in classifying sentences in the **Abstract** sections of his dataset.

Other relevant works focus on an automatic extraction of rhetorical structure of scientific documents to integrate them into end-user applications. Pendar and Cotos [PC08] developed an educational tool for non-native English speakers in post-secondary levels that can classify sentences of an article into one of three rhetorical moves from the C.A.R.S model. An SVM model trained on the **Introduction** sections of 400 published articles from various disciplines and evaluated against 1600 articles with a 7-fold validation resulted in around 80% accuracy with $\kappa = 0.94$ as the upper bound, determined by human inter-annotators agreement. Pendar and Cotos later extended their work to also identify steps of the C.A.R.S model on a larger dataset of journal articles and achieved an average F_1 -measure of 65.4% with a 10-fold cross validation [CP16].

De Ribaupierre and Falquet [dRF17] argued a different perspective on the rhetorical structure of scientific documents. They claim that a document model must take into account the perspective of a user and his information seeking behaviour. In [dRF17], the authors proposed an annotation model with five rhetorical types (see Table 4) that researchers are interested in retrieving from articles, based on a series of interviews with 10 users. They also developed a rule-based system to annotate a corpus of 42 gender studies and psychology journal articles and obtained an average 0.49 F_1 -measure across all rhetorical categories. Finally, they implemented a faceted document retrieval tool built on top of a triplestore that contained the RDF representation of the classified sentences.

4.2.4 Other Disciplines

Although our focus in this dissertation is limited to argumentation mining in Computer Science domain articles, we also review some prominent works in argumentation mining in other disciplines.

The availability of semantically rich, structured resources in the biomedical domain seems to promote rhetorical analysis of its respective articles. HypothesisFinder [MYGHA13] uses machine learning techniques to classify sentences in scientific literature in order to find speculative sentences. Combined with an ontology to find named entities in text, HypothesisFinder can establish hypothetical links between statements and their concepts in a given ontology. The authors’ evaluation of their machine learning approach obtained an F_1 -measure of 0.81 on detecting speculative sentences in biomedical literature.

The Joint Information Systems Committee (JISC), a UK-based non-public body promoting the use of ICT in learning, teaching and research, funded the ART (ARticle preparation Tool) project²⁹ in 2007. The ART project aimed at creating an “*intelligent digital library*”, where the explicit semantics of scientific papers are extracted and stored using an ontology-based annotation tool. The project produced SAPIENT³⁰ (Semantic Annotation of Papers: Interface & ENrichment Tool), a web-based application to help users annotate experiments in scientific papers with a set of General Specific Concepts (GSC) [LS08]. The SAPIENT tool was used to create the ART corpus – a collection of 225 Chemistry scientific papers manually-annotated by experts, which is now freely available under an open-content license.³¹ The development of SAPIENT was eventually succeeded by the SAPIENTA [LSD⁺12] (SAPIENT Automation) tool that uses machine learning techniques to automatically annotate Chemistry papers using the ART corpus as the training model. SAPIENTA’s machine learning approach has achieved an F_1 -measure of 0.76, 0.62 and 0.53 on the automatic detection of Experiments, Background and Models (approaches) from Chemistry papers, respectively.

Ruch et al. [RBC⁺07] treat MEDLINE³² abstracts as “*relatively long documents*” that can be mined to find key sentences containing the ‘gist’ of an article. They used a Naïve Bayes classifier to categorize the Abstract section sentences of biomedical articles into one of their four argumentative moves (see Table 4). In contrast to the above approaches, Ruch et al. neither utilize nor link to any ontological resources in their work.

4.2.5 Discussion

In comparison with the goal of this dissertation, the existing related works in scientific argumentation mining fall short in two aspects: Works like Teufel’s and Anthony’s focus on a statistical, semantics-free analysis of scientific literature with the aim of extracting key sentences from a document, classifying them into a pre-defined schema and then combine them for end-user applications, such as automatic summarization or writing quality assurance. Others, such as SAPIENTA, have ad-hoc

²⁹The ART Project, <http://www.aber.ac.uk/en/cs/research/cb/projects/art/>

³⁰SAPIENT, <http://www.aber.ac.uk/en/cs/research/cb/projects/art/software/>

³¹The ART corpus, <http://www.aber.ac.uk/en/cs/research/cb/projects/art/art-corpus/>

³²MEDLINE, <https://www.nlm.nih.gov/pubs/factsheets/medline.html>

implementations that connect rhetorical blocks of a document to an ontology, custom-tailored for a concrete application. Our work, on the contrary, uses text mining techniques to extract and classify various rhetorical entities on a sentential level and connects the sentences, as well as their contained entities, to a given ontology. Moreover, compared to Groza and de Waard approaches, where the semantic metadata is added to forthcoming articles, our approach can work on any manuscript (published or under development), as long as their full-text is available for machine processing.

4.3 Scholarly Profiling and Recommendation Tools

An important observation from our literature review so far is that, although many of the existing tools and frameworks have an ultimate goal of helping scientists deal with the overwhelming amount of available literature relevant to their tasks, they are completely agnostic of a user’s interests, tasks and background knowledge. In other words, most of these tools have a *content-centric* design, as opposed to a *user-centric* system. One type of user-centric systems are recommender tools that analyze their end-users’ behaviour to predict system items that they may be inclined to read, watch or buy. In this section, we review existing scholarly profiling works and recommender systems that suggest literature to researchers.

4.3.1 Implicit User Profiling

AMiner³³ is a system that combines user profiling and document retrieval techniques [TYZZ10]. General user information, such as affiliation and position, as well as research interests and research networks are presented in textual and visual form. The profiling approach consists of three main steps: profile extraction, author name disambiguation and user interest discovery. *Profile extraction* points to collecting general user information from web pages. Given a scholar’s name, a binary classifier selects web pages according to features, like a person’s name appearing in a page title. All retrieved pages are tagged with categories that are used to generate profile properties, including affiliation, email, address and phone numbers. Extracting research interests are left out in this step, since not all scholars enumerate their interests on their web pages. In addition, research interests should be supported by textual evidence. In a second step, AMiner attempts to link documents with the basic user profiles, in order to obtain a list of a scholar’s publications. Subsequently, AMiner uses publications from different online digital libraries, e.g., DBLP or ACM. To solve the *name disambiguation* problem (i.e., two scholars with the same name), they developed a probabilistic model based on author names. In the final step, they determine *user interests* from the generated linked list of papers. Interests are described based on the detected topics. A topic consists of a mixture

³³AMiner, <https://aminer.org>

of words and probabilities being associated with that word. They propose a probabilistic model called Author-Conference-Topic (ACT) model, where ‘conference’ comprises all kinds of publications, namely journals, conferences and articles. The idea behind this approach is that an author writing a paper uses different words based on her research interests, which denotes the topic distribution. The discovered topic distributions are used as research interests and are stored together with the general information in an extended FOAF format, in what they call a researcher network knowledge base (RNKB). For the evaluation, they utilized pooled relevance judgments [BV04] and human judgments. Seven people rated the retrieved expert lists for 44 topic queries along four expertise levels: definite expertise, expertise, marginal expertise and no expertise. The judges were taught to do the rating according to a guideline following certain criteria, such as how many publication the retrieved scholar actually has for the given topic or how many awards she has received or conferences attended. In a final step, the judgment scores were averaged. In their experiments, they tested different language models along with their ACT model, which was shown to outperform the other models in the best run (P@5: 65.7%, P@10: 45.7%, MAP: 71%).

Generating scholarly profiles has not only been investigated in Information Retrieval, but also in the computational linguistics domain. A first expert profiling approach using Linked Open Data is suggested by [BE08]. They define simple linguistic patterns to identify competences in a user’s research publications. [BB10b] further developed that idea using a GATE pipeline [BB10a] that finds pre-defined skill types in research papers. They define skill types as general domain words that represent theoretical and practical expertise, such as *method*, *algorithm* or *analysis*. Additionally, they applied an adapted TD-IDF filtering algorithm and removed terms from the final list that were considered too broad. In [BKBP12], they extended their system with semantic linking to DBpedia ontology concepts and attempt to find a corresponding concept in the Linked Open Data cloud for each extracted topic. For the evaluation, they conducted a user study with three domain experts, using their own corpus. The users were asked to judge a limited list of 100 ranked topics for a given domain. The list was divided into three sections, *top*, *middle* and *bottom*, and the judges classified the provided topics into *good*, *bad* or *undecided*. Finally, the Kappa statistic was applied to aggregate the three judgments. Overall, 80% of the top ranked topics were marked as *good*.

According to [LPDB10], social media platforms are also widely used among scientists to share research news. Nishioka et al. [NS16] generate scholarly profiles out of social media items, namely Twitter,³⁴ for recommending scientific publications. They examine different factors influencing the recommendation process, such as profiling method, temporal decay (sliding window and exponential decay) and richness of content (full-text and title versus title only). Regarding the profiling method, they took into account the following filtering methods: CF-IDF, an adapted TF-IDF algorithm

³⁴Twitter, <http://www.twitter.com>

using concepts of ontologies instead of full-text terms, HCF-IDF, their own extended hierarchical approach and Latent Dirichlet Allocation (LDA) [BNJ03] topic modeling. For both user tweets and publications, they extract concepts with corresponding labels in the underlying knowledge base through gazetteers. By means of the Stanford Core NLP³⁵ tools, they remove stop words and Twitter hashtags. In their evaluation with 123 participants and around 280,000 scientific publications from economics, they analyzed in total 12 different recommendation strategies, derived as combinations from the three influencing factors and their sub-factors. The participants obtained the top-5 recommendations for each of the 12 strategies and rated the presented publication list on a binary scale. Their results reveal that the most effective strategy was the one with the CF-IDF filtering, the sliding window, and with full-texts and titles. Additionally, it turned out that using titles only in combination with the HCF-IDF filtering produces similarly good recommendations.

Another approach using NLP methods for online profile resolution is proposed by [CSRH13]: They developed a system for analyzing user profiles from heterogenous online resources in order to aggregate them into one unique profile. For this task, they used GATE’s ANNIE plugin [CMB⁺11] and adapted its JAPE grammar rules to disassemble a person’s name into five sub-entities, namely, prefix, suffix, first name, middle name and surname. In addition, a Large Knowledge Base (LKB) Gazetteer was incorporated to extract supplementary city and country values from DBpedia. In their approach, location-related attributes (e.g., ‘*Dublin*’ and ‘*Ireland*’) could be linked to each other based on these semantic extensions, where a string-matching approach would have failed. In their user evaluation, the participants were asked to assess their merged profile on a binary rating scale. More than 80% of the produced profile entries were marked as correct. The results reveal that profile matchers can improve the management of one’s personal information across different social networks and support recommendations of possibly interesting new contacts based on similar preferences.

4.3.2 Scientific Literature Recommender Tools

Papyres [NHA09] is a research paper management system by Naak et al. that comprises a bibliography management and paper recommendation system, based on a hybrid approach. Naak argues that, when locating papers, researchers consider two factors to assess the relevance of a document to their information need, namely, the *content* and *quality* of the paper. He has an interesting view on the definition of the quality of a given research paper: In most collaborative-based recommender systems, items are assigned a single rating value, which is supposed to represent their *overall* quality. However, he argues that such criteria cannot be directly applied to scientific literature, since the rating of a scientific paper can be relative to the objective of the researcher. For example, a

³⁵Stanford Core NLP, <http://stanfordnlp.github.io/CoreNLP/>

researcher who is looking for implementation details of an innovative approach is interested mostly in the implementation section of an article and will give a higher ranking to documents with detailed information, rather than related documents with modest implementation details and more theoretical contributions. This does not necessarily mean that the lower ranking documents have an overall lower quality. The Papyres system targets this issue by allowing researchers to rank multiple criteria of a document, similar to the criteria of most peer-review processes, e.g., Originality, Readability, Technical Quality. Based on these multi-criteria, Papyres then calculates an overall rating for a collaborative filtering of articles. However, the content-based filtering of Papyres does not actually deal with the content of the paper, rather it is based on the metadata and bibliographical characteristics of the document. The authors of Papyres evaluated their system with a set of 100 different research-paper pairs in a user study, where the participants provide explicit ratings for papers and obtained a Mean Absolute Error of 17% using their best performing approach.

Gipp et al. [GBH09] criticizes content-based recommendation approaches in the context of scientific literature. He states that text-based analysis of research literature is problematic since it has to deal with unclear nomenclatures and semantic relations, like synonymy and hypernymy. He also reprehends citation-based recommender systems and debates that based on classic references, it can only be determined that two documents are *somehow* related. Instead, he introduces Scienstein [GBH09], a holistic paper recommendation approach that combines citation, ranking, and impact factor with text mining of collaborative links (expressing how documents are related) and annotations that users create.

CiteSeer [BLG00] is a digital library that provides personalized filtering of new publications, based on a user's browsing history, and recommendation of interesting research trends, citations, keywords and authors to its end-users. CiteSeer uses machine learning techniques to extract bibliographic information from scientific literature and create a citation graph. In response to users' queries, CiteSeer ranks papers by the number of times they are cited in its database. CiteSeer represents a user's interest as a set of *pseudo-documents* that contain a set of features automatically extracted from the literature that the user browsed to in the system, including keywords, URLs and word vectors. These documents are assigned a weight that corresponds to their influence and serve as indicators of the user's interest in its recommendation approach. CiteSeer allows users to manually adjust the weight of a pseudo-document in their profile. The relatedness of documents in CiteSeer is calculated based on common citations and their inverse document frequency in the given instances. Bollacker et al. later redesigned the CiteSeer architecture for scalability and renamed their system to CiteSeer^X.³⁶

³⁶CiteSeer^X, <http://citeseerx.ist.psu.edu>

Similarly, Google Scholar³⁷ is a dedicated search engine with a wide coverage of scientific literature in multiple domains and disciplines. All researchers with published works can have a user profile on Google Scholar. The recommendation engine of Google Scholar then analyzes the content of works within the profile of each user and combines them with a graph-based analysis of their citation network in order to recommend articles to its users.

QuickStep by Middleton et al. [MSDR04] uses ontology-aware recommendation of scientific literature, based on a taxonomy of topics from the Computer Science domain. The authors performed several studies to evaluate the effect of integrating topic ontologies on the performance of recommender systems. Their results show that users of the ontology-aware version of their system were 10% “happier” (i.e., more satisfied with the recommendation of the system) than the conventional filtering approaches, since users received a broader range of interesting documents, than the ones explicitly defined in their profiles. However, their content-based approach merely matches the similarity of a user’s interests versus the topics of a document detected by a classifier and does not analyze the actual content of the paper.

Semantic Scholar³⁸ is a “*smart*” search engine started in 2015 for journal articles, developed by the Allen Institute for AI³⁹, that combines NLP and computer vision techniques to quickly find a survey paper or identifying key literature for citation. Initial versions of Semantic Scholar covered computer science domain articles, though as of 2017 it has added biomedical journal articles to its already massive index.

Another academia-industry collaborative project is Dr. Inventor [RS15], a European Commission’s Seventh Framework-funded (EU FP7) project⁴⁰ (2014-2016) that aims at creating a “*personal research assistant, utilizing machine-empowered search and computation... [to help researchers] by assessing the novelty of research ideas and suggestions of new concepts and workflows.*” The project has received more than 2.6 million Euros and involved multiple university and research institutions from Germany, Spain, Ireland, UK, and Czech Republic. Relevant to the Dr. Inventor project, O’Donoghue et al. [OPO⁺14] introduce a four-level hierarchy of computational process for sustainable, computationally creative systems that can produce “*new ideas in the form of knowledge or artefacts that represent that knowledge.*” They demonstrate two applications that, given an input and a goal, can produce *creative* artefacts and processes.

³⁷Google Scholar, <https://scholar.google.com>

³⁸Semantic Scholar, <https://www.semanticscholar.org>

³⁹Allen Institute for AI, <http://allenai.org>

⁴⁰Dr. Inventor Project, <http://drinventor.eu>

4.3.3 Discussion

Existing recommender systems are showcases of how, simply by analyzing documents’ metadata, novel, efficient methods of literature retrieval can be provided to researchers, in order to help them find interesting documents from the mass of available information. However, in most cases, the documents are merely recommended based on their bibliographical information or citation networks and not based on their actual content (which is what matters the most to researchers). Such brittle recommendation mechanisms fall short in two ways: (i) the recommender algorithms will be crippled by the absence of such rich metadata and (ii) they do not account for the background knowledge of the researchers, i.e., what they already know.

Albeit very similar in its outlook to create personal research assistants, Dr. Inventor project’s focus is to “*promoting scientific creativity by [using] web-based research objects*”, specifically for the Computer Graphics domain researchers. Interestingly, they conducted a survey of researchers on their habits in reading and finding research articles [CYY14] that outlined finding, reading and comparing the rhetorics of different articles with their research goals as the most difficult and time-consuming tasks, which we target to facilitate in this dissertation.

4.4 Summary

In this chapter, we reviewed existing research in areas related to our work in this dissertation. We reviewed the efforts regarding semantic modeling of scholarly artifacts, manual and automated approaches in argumentation detection in scientific literature and existing academic recommender systems. At the end of each section, we briefly discussed how our personal research agents are different from similar existing works.

Chapter 5

Personal Research Agents Design

The ultimate goal of constructing a knowledge base is to enable the agent to realize its tasks, through automatically fulfilling the users' information needs. Developing the knowledge base's underlying ontology, such that it has sufficient expressivity and flexibility to adapt to various domains' content, is a complex task. In this chapter, we explain our methodology for scholarly knowledge base construction and the design decisions we made to meet the requirements described in Chapter 2.

5.1 An Abstraction Model for the Knowledge Base

A knowledge engineering process encompasses activities conducted for the acquisition, modeling and application of knowledge elicited from diverse, unstructured resources. Here, we formulate an end-to-end, flexible workflow for scholarly knowledge engineering. As illustrated in Figure 8, the input to the workflow is a set of scientific literature in a natural language, like English. The output of our workflow is a knowledge base, populated with pertinent information mined from input documents, including literature relevant to the agent's tasks or publications provided by the user for profiling purposes. The decision as to what information must be stored in the knowledge base is derived from enumerating concepts and relations required to satisfy the requirements explained in Section 2.1. The following sections describe how we designed our knowledge base as a collection of inter-connected semantic models and the phases of our methodology in more detail.

5.1.1 Competency Question-Based Ontology Construction

Based on ideas borrowed from Test-driven Software Development techniques, our design of the knowledge base starts from postulating the types of queries that the knowledge base has to answer in order to fulfill the agent's services. One approach introduced by Ren et al. [RPM⁺14] suggests

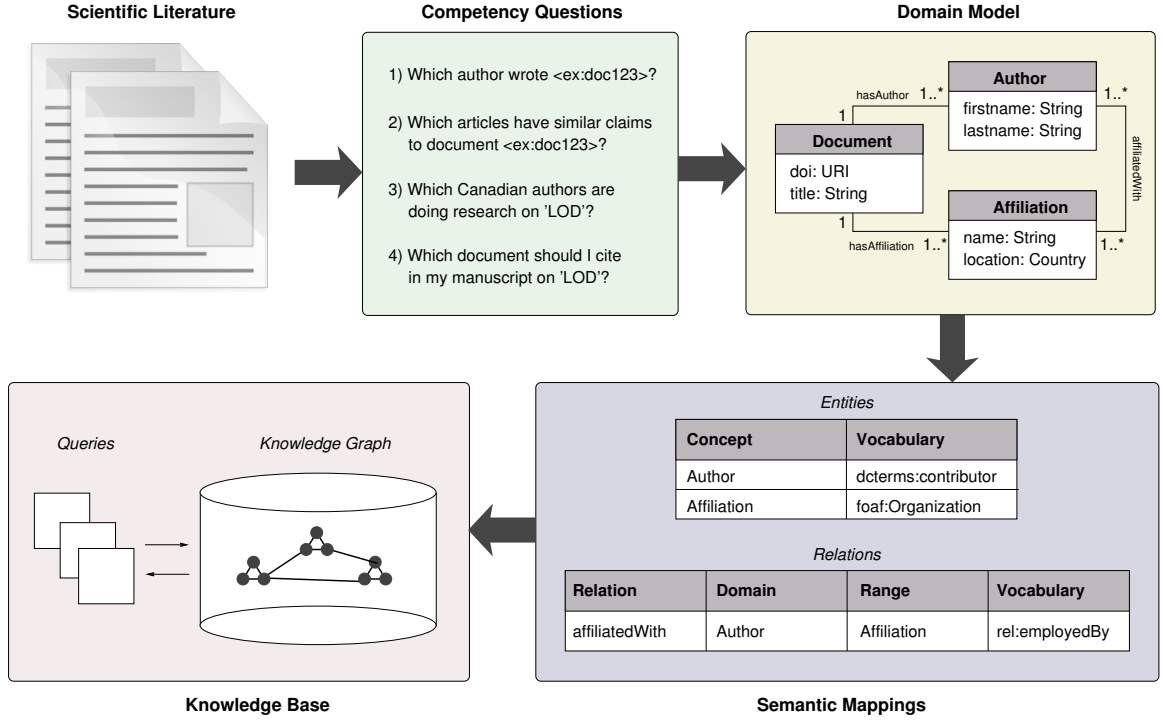


Figure 8: A flexible workflow for scholarly knowledge base construction

to derive the knowledge base model from a set of *competency questions*. A Competency Question (CQ) [UG96] is a natural language sentence that expresses patterns for types of questions a user wants to ask from a knowledge base. Ren et al. also showed that most competency questions can be categorized into at most 12 archetypes of generic patterns. Each pattern is a template with variables that will be filled at query presentation time with user-defined values. Going back to the functional requirements we iterated in Section 2.1.1, we collected a list of competency questions corresponding to each requirement (see Appendix D for the complete list). We then refactored the questions into more generic patterns, removed the duplicates and proposed the archetypes shown in Table 5. Note that more complex queries can be concocted by combining two or more archetypes, or through a reification process.

The placeholders in CQs serve two essential purposes: First, possible values of the placeholders in CQ archetypes determine the types of entities that must be stored in the knowledge base. When referring to entities by their type, the placeholders represent the possible *classes* of things in the underlying ontology, referred to as *class expressions* (CE), e.g., the class of ‘documents’, ‘persons’, or ‘countries’. *Individuals* (I) are specific instances of classes that bear a set of attributes and can be uniquely identified in the knowledge base, e.g., a document identified by <ex:doc123>. While *datatype properties* (DP) attach literal values such as strings and numerics to individuals, *object*

Table 5: Archetypes of the knowledge base competency questions (derived from [RPM⁺14])

Competency Question Pattern	Example	Req.
Which [CE] [OP] [I]?	<ul style="list-style-type: none"> • Which authors wrote <ex:doc123>? • Which organizations are affiliated with <ex:author123>? • Which articles are similar to <ex:doc123>? 	R5 R5 R2, R3
Which [CE] [OP] [CE]?	<ul style="list-style-type: none"> • Which documents have a claim? 	R1, R2, R3
Which [CE] has [I]?	<ul style="list-style-type: none"> • Which articles have (mention) <dbp:prototype>? 	R2, R3
Which [CE] should I [OP]?	<ul style="list-style-type: none"> • What topics should I learn? • What documents should I read? 	R4 R3, R6
Which [CE] published [QM] articles?	<ul style="list-style-type: none"> • Which country published the highest number of articles? 	R5
How many [CE] [OP] [I]?	<ul style="list-style-type: none"> • How many journals are cited in <ex:doc123>? • How many articles are authored by <dbp:concordia_university>? 	R1, R5 R5
Which [CE] has [QM] [DP]?	<ul style="list-style-type: none"> • Which article has the highest number of pages? 	R1, R3

Legend: CE = Class expression, I = Individual, OP = Object property, DP = Datatype property, QM = Quantity modifier

properties (OP) represent arbitrary relations between classes or individuals in the ontology, such as an authorship relation between a document and a person. The difference between the two property types is that, by dereferencing the object of an object property, we obtain a class or an individual, rather than a typed value. Furthermore, the CQs will also imply restrictions on properties, such as cardinality constraints that the ontology design must satisfy. For example, the ontology must allow multiple authors to be connected with a document but only one title for each article. The collective set of classes of things related to the scholarly domain and plausible relations between their instances would shape the representation of the agent’s knowledge base *schema*.

Second, since most CQs have relatively simple syntactic patterns, as Zemmouchi-Ghomari and Ghomari investigated in [ZGG13], they can be directly translated into SPARQL queries over the knowledge base. Not only can we then incorporate the queries into the agent’s service implementations, but the queries will also play the role of unit tests that check for consistency and completeness of the knowledge base ontology.

5.1.2 Domain Model

Possible values of the class expressions in CQs define the entity types that should be modelled and persisted in the knowledge base. Together with the attributes of individuals and their inter-relationships, they describe the facts and assumptions about the world we attempt to represent in

the knowledge base, captured in a *domain model*. Constructing the domain model follows standard knowledge base construction techniques, such as the models from the NIST Automatic Content Extraction¹ (ACE) and Text Analysis Conference² (TAC) series. The domain model in our workflow, illustrated in Figure 9, encompasses three types of information:

Entities are concepts required to answer the competency questions (i.e., class expressions and individuals). Entities model real-world things like people and universities, but they can also reflect abstract concepts like a **Contribution** sentence or an algorithm.

Mentions are spans of text that refer to entity individuals. Mentions can be uniquely identified using their offsets in documents, which index their start and end characters in a text. However, one mention may be linked to several entity types, e.g., mention of a person name in the document metadata can be linked to both `<foaf:Person>` and `<ex:bahar>`, allowing for overlapping entities in documents.

Relations represent the associations between two or more entities in the knowledge base. Each relation may optionally also represent *how* the constituents are related to each other. For example, relation *mentionedIn*(t, d) represents that a topic t is mentioned within a document d 's text, but $(\text{mentionedIn}(a, d) \wedge \text{authorOf}(a, d))$ implies that a has written document d . Relations may optionally have a textual mention that stands as the provenance data for how the relation was extracted. Otherwise, the mention can be omitted if the relation was automatically inferred.

5.1.3 Semantic Mappings

Answering competency questions like ‘Which two documents have similar contributions?’ or ‘What countries publish the highest number of articles?’ requires access to a vast amount of machine-readable knowledge, encompassing both common-sense and domain-specific facts. This brings us to the next phase of our knowledge base construction workflow, which focuses on constructing a semantic representation of the knowledge base entities. A semantic representation can be thought of as a mapping of domain model concepts on various abstraction levels, to classes and individuals in existing knowledge bases or shared vocabularies.

On a high level, we regard every piece of information extracted from documents as an *annotation*. Annotations are essentially a set of markups that associate globally meaningful metadata to a document. Annotations are non-linguistic, syntactical structures that can attach to a document as a whole (e.g., a document identifier) or to specific spans within their textual content (e.g., a university

¹Automatic Content Extraction, <https://www ldc.upenn.edu/collaborations/past-projects/ace>

²Text Analysis Conference, <https://tac.nist.gov/>

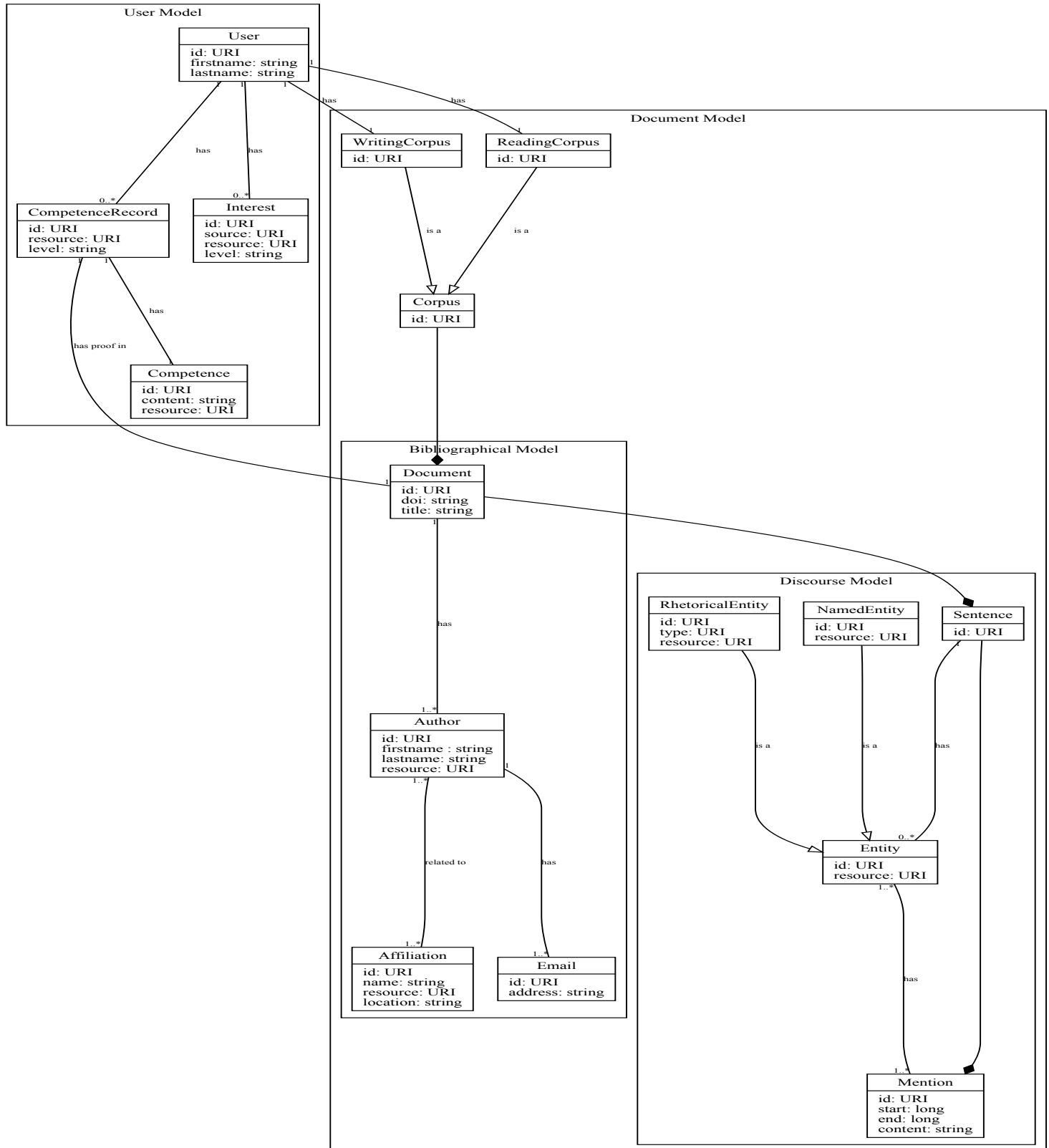


Figure 9: The agent's knowledge base domain model

Table 6: High-level semantic mapping of our PUBO domain model. The vocabulary namespaces used in the table can be dereferenced using the URIs shown in Appendix C.

Domain Concept	Description	Linked Open Term
Corpus	A corpus is a collection of documents.	pubo:Corpus
ReadingCorpus	A corpus that contains documents that the user has read.	pubo:ReadingCorpus
WritingCorpus	A corpus that contains documents (co-)authored by the user.	pubo:WritingCorpus
Document	A scholarly manuscript (see Definition 5.2.1).	bibo:Document
User	A (human) end-user that interacts with the agent.	um:User

Property	Type	Domain	Range	Linked Open Term
id	object	rdfs:Resource	xsd:anyURI	rdfs:Resource
type	object	rdfs:Resource	rdfs:Class	rdf:type
source	object	rdfs:Resource	xsd:anyURI	rdfs:isDefinedBy

name). Annotations are used as implications of entities, mentions and relations in the knowledge base and may optionally carry further information, such as their confidence level or provenance information, especially in the agent’s automated settings. We designed our *PUBlication Ontology* (PUBO)³ as a high-level schema for describing scholarly literature and their associated annotations that reuses and links to several open vocabularies (see Section 4.1.1). Table 6 shows the PUBO mappings that remain unchanged, regardless of what domain the workflow is applied on.

On the concept level, semantic mappings are provided by end-users using a declarative language for ultimate customizability. Say, if an end-user has access to an existing knowledge base of scholarly information, in which all authors are instances of a `<my:Author>` class, he can then declare a semantic mapping of author names extracted from documents to be mapped to his class of choice in the external knowledge base. This way, the knowledge graph created by the agent can be interlinked and integrated with additional information not directly available in the document’s text.

It should be noted that the terms and vocabularies referenced in semantic mappings will be directly integrated in the knowledge base and therefore, affect the structure of the corresponding SPARQL queries used in the agent’s services. This design decision provides great flexibility in terms of dynamically customizing the knowledge base model, at the expense of high coupling between its underlying ontology and service implementations.

³PUBlication Ontology (PUBO), <http://lod.semanticsoftware.info/pubo/pubo#>

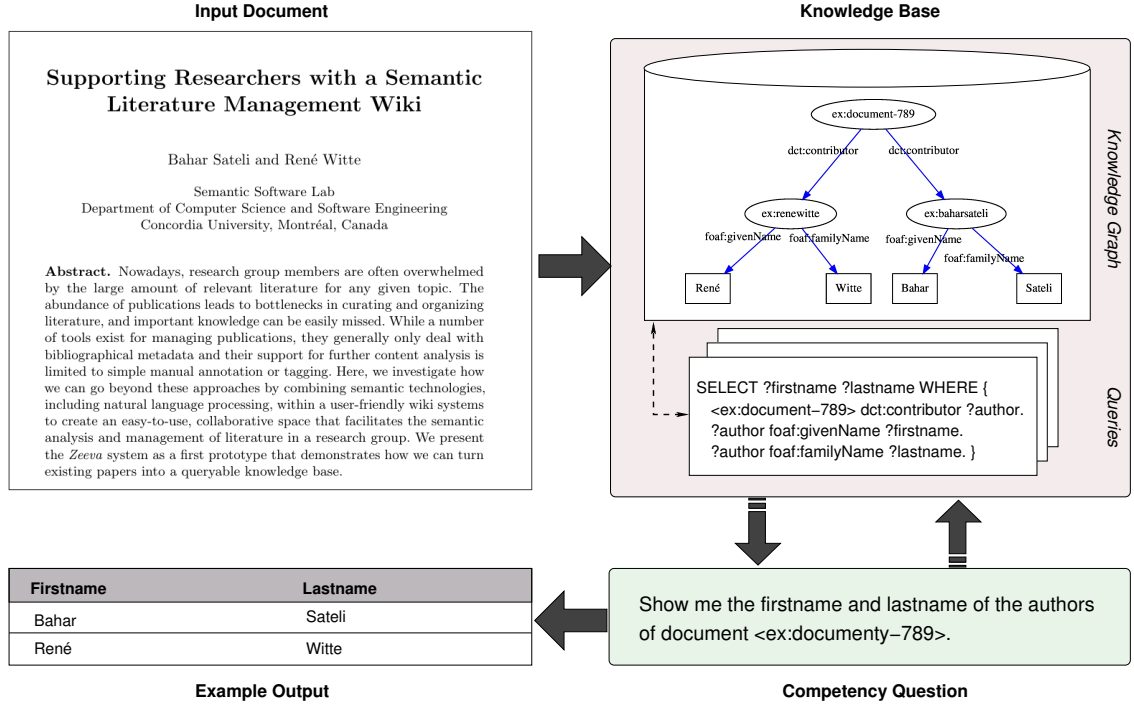


Figure 10: Example processing using our workflow with its input and output

5.1.4 The Knowledge Base

With the description of the knowledge engineering components in place, we can now define a formalism for a graph-based representation of our agent’s knowledge base. The agent’s knowledge base is defined as graph $G = (V, R)$. The graph nodes $V = \{E \cup M\}$ is the union of entity annotations E describing domain concepts, like *foafPerson(bahar)*, and mention annotations M , which are the textual proof of entities in documents. The knowledge graph’s edges R form a set of semantic relations, between high-level concepts, such as *hasDocument(corpus, document)* between two classes, as well as entity-entity associations, like *hasAuthor(thesis, bahar)*.

In addition to the knowledge graph, the agent’s knowledge base contains a semantic description of its services and their corresponding resources, like the SPARQL queries translated from the competency questions explained in Section 5.1.1. Figure 10 shows an example processing workflow using our knowledge base construction methodology.

5.2 Semantic Modeling of Scholarly Literature

After establishing the overall methodology used to populate a scholarly knowledge base, we begin this section by enumerating the domain model entities we are interested in extracting from scholarly documents and explain our approach in formally representing them using semantic web technologies.

We have liberally used the term ‘document’ so far, but before proceeding with a formal, abstract model of scientific literature, it is important to first establish a consistent definition:

Definition 5.2.1. (*Document*). A *document* is any heterogeneous scholarly inscription that contains an informative description of its authors’ scientific work, such as a conference proceedings, journal article or academic dissertation.

Definition 5.2.1 is comprehensive enough to include various types of scholarly documents without any hard restrictions on the length, formatting, typesetting or peer-review status of the manuscript. Furthermore, the choice of medium through which the document was manufactured (PDF, HTML, OCR’ed document) is inconsequential in our research, as long as the full-text content of the document is machine-readable and can be accessed by the agent. Based upon this definition, we describe how we model the structure and semantics of a scholarly document with a graph-based representation, using the Resource Description Framework (see Section 3.3).

5.2.1 Bibliographical Metadata

Uniquely identifying and collating documents and their contributors across the publishing landscape depends on the availability of scholarly literatures’ bibliographical metadata. Such information is added to documents using various markup languages. These specific languages intermingle a set of pre-defined annotations and content together to flag structural and semantical components of a document [Ril17]. Based on their use cases, we particularly focus our design on two types of metadata, which are needed for fulfilling our requirements:

Descriptive metadata is needed for the management and categorical retrieval of documents (Requirements #1–3, 5–6). Vocabularies for terms, such as title, authors, affiliations, subjects, and publication dates, are examples of descriptive markup.

Structural metadata comprises vocabularies for navigation and fine-grained retrieval of documents’ implicit knowledge, e.g., only retrieving the **Contribution** sentences of an article or displaying the **Results** section (Requirements #1–3).

Earlier in Section 4.1.1, we reviewed some of the existing controlled vocabularies for descriptive bibliographical entities in a document. Following the best practices of creating linked open datasets [HB11], we reused terms from existing vocabularies to the extent possible and only augmented the PUBO schema with additional classes and properties when no existing suitable terms were available. Table 7 shows the classes and properties we added to the PUBO schema.

For structural modeling of documents, we follow the schematic structure of computer science articles as investigated by [HP10] to decompose the full-text content of each article into non-overlapping

Table 7: Bibliographical metadata vocabularies in our PUBO schema. The vocabulary namespaces used in the table can be dereferenced using the URIs shown in Appendix C.

Class	Description	LOV Term
pubo:Author	An author is a person Entity, who has contributed to the document under study in some capacity. There is no pre-defined format for author names, but often metadata contains the full-form of each author's name.	dcterms:contributor
pubo:Affiliation	An affiliation is typically an organization or academic body that authors of a document are associated with. Affiliation information often contains both the name of the organization and its geographical metadata, such as the country and city where it is located.	foaf:Organization
pubo:Abstract	An abstract is a relatively short paragraph that contains a condensed version of the contributions of an article with the purpose of helping readers quickly ascertain the publication's purpose.	doco:Abstract
pubo:Section	A section is a division of a document's text into smaller logical constituents.	doco:Section

Property	Type	Domain	Range	LOV Term
firstname	datatype	pubo:Author	xsd:string	foaf:givenName
lastname	datatype	pubo:Author	xsd:string	foaf:familyName
email	datatype	pubo:Author	xsd:string	vcard:hasEmail
name	datatype	pubo:Affiliation	xsd:string	foaf:name
location	datatype	pubo:Affiliation	xsd:string	geo:locatedIn
title	datatype	pubo:Document	xsd:string	dce:title

sections. Unlike for other natural sciences disciplines, Hyppönen and Paganuzzi [HP10] found that computer science articles often deviate from de-facto standards like IMRAD [NN14], C.A.R.S [Swa90] or ABCDE [dWT06]. Therefore, there exists no universal structure for our domain's documents, according to which the body of a document can be easily segmented. We can only decompose a document based on the existence of section headers. We further classify each section into a semantic category available in the DoCO ontology, like `<doco:Bibliography>` for the References of an article. If no suitable class is found, we merely classify it as an instance of the `<doco:Section>` class. The RDF graph in Figure 11 illustrates our semantic model of the descriptive and structural entities of the example document shown in Figure 10, using our PUBO schema.

5.2.2 Scientific Discourse Modeling

Requirements, like automatic summarization (Requirement #1) and finding related work (Requirement #3), require an understanding of the *content* of each document by the agent. For instance,

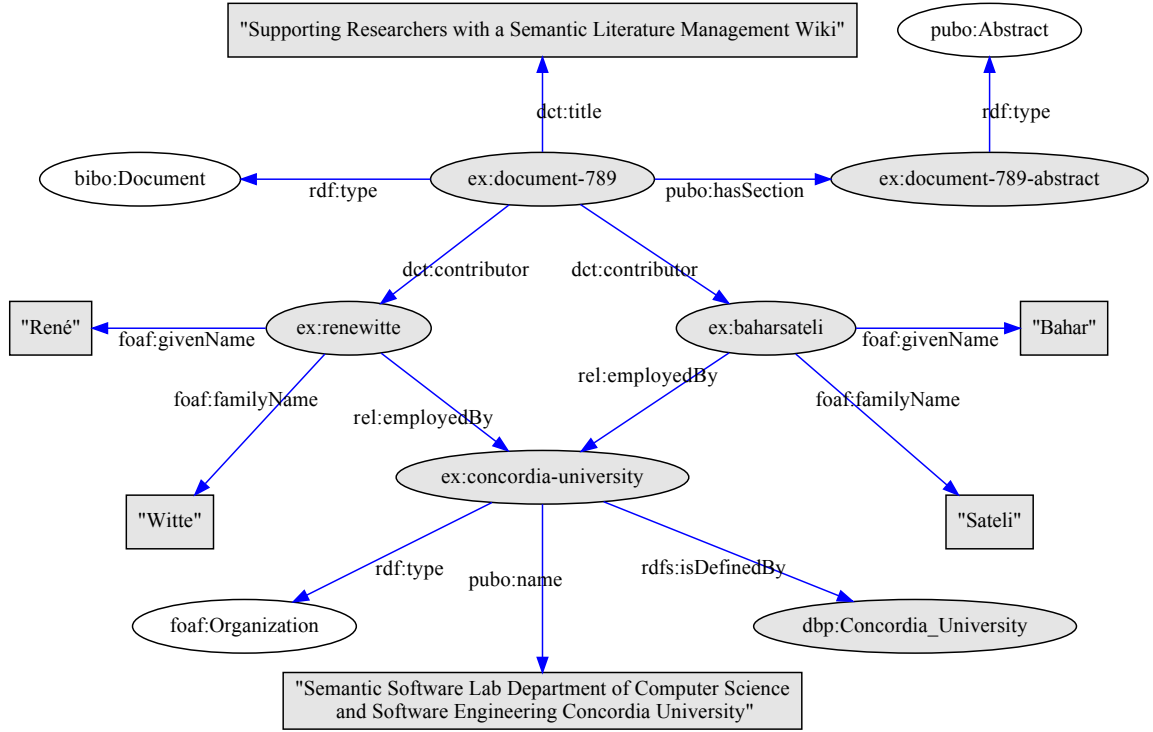


Figure 11: The agent's semantic model of bibliographical entities in a document

merely matching author and affiliation metadata is ineffective in finding similar work to a given manuscript. Rather, the agent needs to understand the *meaning* of each document and the rhetorics of its authors to determine its pertinence for a task. Here, we describe how we capture a semantic model of scholarly literature, such that it can convey the authors' rhetorical moves in a machine-readable format (Requirement #8).

Representing Argumentations with Rhetorical Entities

We previously illustrated in Chapter 3 that scholarly documents are not monolithic structures, rather they are composed of several cohesive rhetorical moves to establish and support scientific argumentations. We exploit this intrinsic characteristic to model the meaning of a document as a set of *Rhetorical Entities* (REs), such as the ones discussed in Section 4.1.1. To this end, we examine each sentence within the body of a document in order to determine whether it represents a rhetorical move.

Definition 5.2.2. (*Rhetorical Entity*). A rhetorical entity is a sentence in a document that presents a scientific argumentation move, such as establishing a niche or announcing research findings.

In this dissertation, we selected two rhetorical entity types to detect from scholarly documents, which are sufficient to satisfy our functional requirements. Note that other rhetorical entity types can be detected using a similar approach:

Contributions are sentences in a document that provide an overview of the authors’ work and outline the main purpose of the document. Contributions are rhetorical moves by authors to occupy a niche they established in their respective research space (similar to C.A.R.S model Move 3, Step 1 [Swa90]). Contribution sentences of a document can be combined into an extractive summary or serve as a comparison criterion for the semantic similarity of two distinct works.

Claims are sentences in a document that announce the principal research findings of its authors (similar to C.A.R.S model Move 3, Step 2 [Swa90]). In our design, claim sentences are those which (i) are a statement in form of a factual implication, and (ii) have a comparative voice or assert a property of the author’s contribution, like novelty or performance superiority. Claim sentences can be used to distinguish two similar works, for example by identifying authors who claim novelty in their approach.

To formulate a set of patterns with which we can identify rhetorical entities in a given document, we performed an extensive study of computer science research articles from the software engineering and computational linguistics domains and curated grammatical structures and discourse markers that indicate the possible presence of a rhetorical entity and further aid in its classification. We constructed several lists of terms found within lexical and grammatical structures of rhetorical entities in the AZ-II [TSB09], Mover [AL03] and ART⁴ gold standard corpora, including:

Rhetorical Moves, which are a set of verbs and verb phrases that indicate scholarly research activities, such as ‘investigate’, ‘implement’, ‘verify’, or ‘examine’. The verbs in this list are further classified into cohesive sub-lists, such as the list of verbs to indicate an *Action* (e.g., ‘develop’), as opposed to verbs used for *Presentation* (e.g., ‘demonstrate’) of a research finding.

Discourse Cues, which are signal words and phrases that authors use to refer to their presented work or parts thereof, relative to the sentence where the cues are mentioned. Phrases like ‘in this work’ or ‘in what follows’ are examples of discourse-level deictic terms.

Domain Concepts, which is a list of domain-dependent terms found in computer science articles, like ‘framework’, ‘algorithm’, ‘approach’, ‘system’, or ‘article’. When referring to their contributions within the research space, authors frequently use domain concepts in their rhetorical moves.

⁴The ART Project, <http://www.aber.ac.uk/en/cs/research/cb/projects/art/>

Table 8: High-level semantic mapping of our PUBO argumentation model. The vocabulary namespaces used in the table can be dereferenced using the URIs shown in Appendix C.

Domain Concept	Description	Linked Open Term
Sentence	A sequence of words and symbols in a document.	doco:Sentence
Entity	A real-world or abstract concept in the knowledge base.	pubo:Entity
NamedEntity	A type of Entity that refers to a domain topic.	pubo:LinkedNamedEntity
RhetoricalEntity	A type of Entity that represents the stance of a document’s author.	sro:RhetoricalElement
Mention	A span of text in a document that refers to an entity.	doco:TextChunk

Property	Type	Domain	Range	Linked Open Term
startOffset	datatype	Mention	xsd:long	oa:start
endOffset	datatype	Mention	xsd:long	oa:end
content	datatype	Mention	xsd:string	char:cnt

We mark up every matching occurrence of the trigger terms in our lists against the sentences in the textual body of a given document. Based on the rhetorical verb used in each sentence, we categorize it into a rhetorical type (e.g., **Contribution** or **Claim**). Using the PUBO schema, each annotated sentence becomes an instance of the `<sro:RhetoricalElement>` class. The span of text covered by the rhetorical entity becomes an instance of the `<pubo:Mention>` class that stands as the provenance of where that sentence was found in a document. To precisely locate where a mention is found, we retain the start and end offset of the individual mention in the knowledge base, by counting the index of the first and last character of the sentence from the beginning of the document (counting up from zero). Maintaining the anchor offsets of rhetorical entities not only allows us to find their textual content, but can also be used for retrieving adjacent sentences on demand from the original text that could provide contextual information for the readers’ comprehension (e.g., in the case of co-references). Table 8 shows the vocabularies in our PUBO schema for the semantic modeling of rhetorical entities.

We use the example document shown in Figure 10 to create an RDF graph of one of its **Contribution** sentences: Figure 12 shows a rhetorical entity instance and its mention individual connected with the document instance in the graph. Note that the light nodes in the graph are the schema (classes) used in our model and the shaded nodes are individual instances in the knowledge base. We reused a selected set of vocabularies from the DoCO, OA and SRO ontologies (see Section 4.1.1), in addition to our PUBO schema.

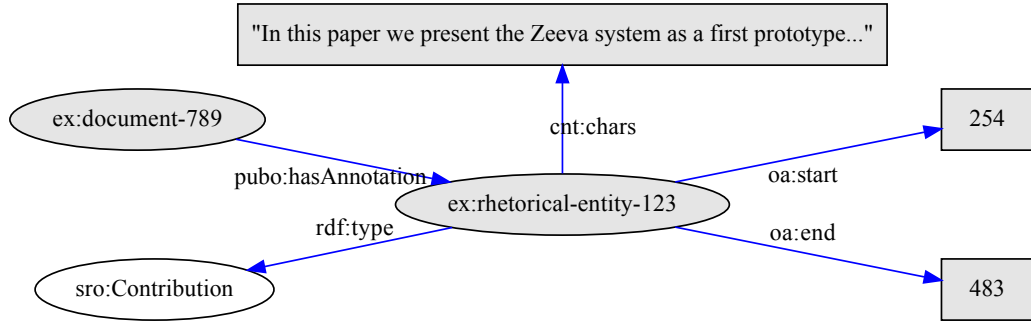


Figure 12: Agent’s model of relations between a document and a rhetorical entity

Document Topics Representing a Document’s Meaning

Rhetorical entities enable our agent’s services, like summarizing a document, by retrieving its *Claims* and *Contributions*, but they cannot answer what precisely a rhetorical sentence is *about*. Traditionally, so-called *topic models* [BL09] were created from a collection of documents that would give a sense of which topics a document is about, as well as the correlations and trends between the topics. Topic models are probabilistic models that aim at discovering patterns of words reuse in documents, typically based on a hierarchical Bayesian analysis of their text. The idea is to connect ‘similar’ documents based on the distribution of topics within their content. However, these probabilistic models do not take into account the linguistic features of the documents’ content: The only language feature used in probabilistic models is stopwords removal, which is used for dimension reduction of the underlying model, rather than analyzing the text for semantics. Moreover, in probabilistic topic modeling algorithms, like Latent Dirichlet Allocation (LDA) [BNJ03], the size of the vocabulary from which topics are selected are known a priori. However, the automated working context of our agent, the size and growth rate of relevant publications, as well as the diversity of topics in the computer science domain literature, precludes a manual construction of such a vocabulary.

In our design, we take a different approach on topic modeling: We reuse the existing resources on the linked open data cloud (see Section 3.3) as the possible vocabulary for topics within an article. This approach has two advantages: *(i)* the LOD cloud is an ever-growing repository with extensive coverage of both domain-dependent and common-sense knowledge, and *(ii)* the resources in the linked open datasets are already available as machine-readable data. Hence, to model the *meaning* of REs (and the documents in general) for the agent, we augment the documents’ sentences with domain topics by comparing their words against existing resources on the LOD cloud. From a linguistic point of view, we readily know that all domain topics, like names of frameworks, tools, or algorithms, are expressed as nouns and noun phrases. Therefore, given the syntactical parsing of the sentence underlying a rhetorical entity, we can largely reduce the topics space to those overlapping

a noun or noun phrase. This approach is referred to as *named entity recognition* (NER), where mentions of entities in a text are located and grounded (linked) to existing semantic categories or resources in a knowledge base. With this design in mind, we can now provide a definition for domain topics in scholarly documents:

Definition 5.2.3. (*Named Entity*). A named entity is a noun or noun phrase within a document’s content that refers to any entity that can be denoted with a proper name, including physical objects like persons or locations, as well as abstract concepts like algorithms or measurement units.

An important issue to address in the semantic modeling of named entities is separating the *surface form* of an entity from its *meaning*. The surface form of an entity is a linear sequence of characters covering the words or symbols that comprise the name of the entity. The meaning of an entity, on the other hand, is the additional data available when the corresponding LOD resource is dereferenced. For example, ‘*linked open data*’, ‘*linked data*’ and ‘*LOD*’, although different in surface form, likely refer to the same concept within an article. Therefore, all three phrases must be linked to the same LOD resource, for instance, <dbpedia:Linked_data> in the DBpedia ontology. The named entity modeling is conducted in three steps:

Spotting is finding words in a document that refer to a named entity. Only the surface forms for which there exists a resource on the LOD cloud must be marked up for semantic analysis.

Disambiguation is finding the right *sense* for the named entity. For example, the word ‘*tree*’ in a computer science article most likely refers to the data structure, rather than a botanical organism.

Grounding is linking the surface form to the corresponding LOD resource, using a uniform resource identifier (URI).

Subsequent to aligning the text with LOD resources, every span covering a named entity becomes an annotation, which eventually translates into an RDF triple representing the document’s topic in the agent’s knowledge base. The annotation itself is the subject of the triple, using <pubo:Mention> as its type. The textual content of the named entity is stored as a literal value, connected to the subject using <cnt:chars> as the predicate. By inheriting the <pubo:Entity> class, the boundaries of the named entity annotation are stored using their start and end offsets in a text. Figure 13 shows example triples, representing the named entities in our running example document. We use the <rdfs:isDefinedBy> predicate to connect each mention with an LOD source, e.g., <dbpedia:Linked_data>. This way, we can find all the unique topics of a document by dereferencing their source URI. At the same time, the mention triples referring to the same URI can be used to

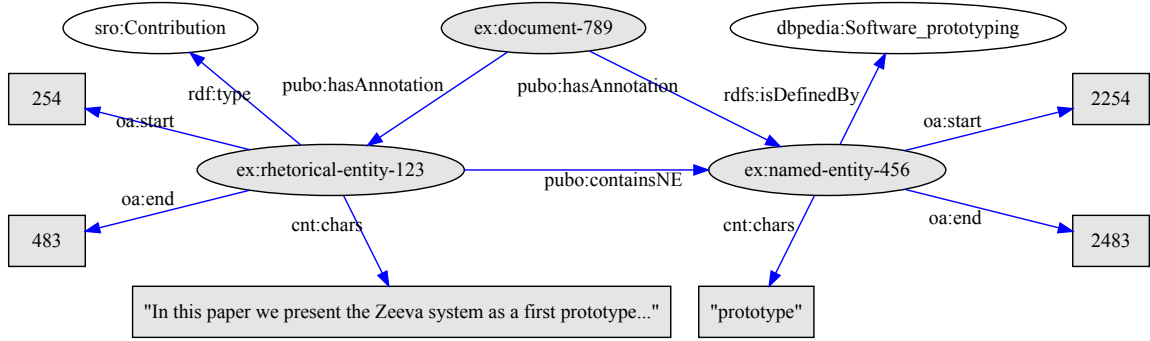


Figure 13: Agent's model of named entities in a document

find their raw frequency in text. This design is reflected in Figure 13 using the `<rdfs:isDefinedBy>` predicate between the named entity individual and `<dbpedia:Software_prototyping>`.

A rather interesting question here is whether all of the detected named entities are representative of the document's topics, or if entities in certain regions of the documents are better candidates. To test this hypothesis, we further annotate each named entity individual with whether it falls within the boundary of a rhetorical entity. We will later evaluate this hypothesis in Chapter 8.

5.3 Semantic Modeling of Scholarly Users

Similar to document modeling, we now continue to establish a formal model for the agent's end-users. Our goal here is to create a *semantic profile* for each end-user, based on characteristics, like knowledge and interests of a user, as we explained earlier in Section 3.4.

Before defining our scholarly user model, we first need to make an observation with regard to profile features across two aspects: (i) change over time and (ii) the acquisition process: User features like knowledge level, interest topics and goals are attributes that change over time. They may even change during each session that a user is interacting with the system. In fact, there is an implicit affiliation between a task, a set of interests and the knowledge level of a user: In order to successfully conduct a task (e.g., writing an article about topic t), the user is required to possess a certain level of knowledge (of topic t). Therefore, as the user conducts various tasks in the system – dictating his relevant interest topics – his knowledge level varies. On the other hand, other features, such as background information or individual traits of a user are relatively stable (at least within one session) and may moderately change over long periods of time.

We can also distinguish user features considering the manner through which meaningful information is acquired. Populating a user profile with related information can be conducted in two fashions: (i) in an explicit way, by directly asking users to provide information to the system, or

Table 9: Scholarly user profiles features

Feature	Adaptation			Acquisition	
	Mostly Stable	Change Over Time	Change Within a Session	Implicit	Explicit
Knowledge	-	✓	-	✓	-
Interests	-	✓	✓	✓	✓
Goals/Tasks	-	✓	✓	-	✓
Background	✓	✓	-	-	✓
Individual Traits	✓	✓	-	✓	✓

(ii) implicitly, by observing the user’s behaviour (e.g., session duration, user’s click-trail) or finding external resources about the user (e.g., the user’s social media profiles).

In our work, we focus our scholarly user profiling on capturing the users’ knowledge, interests, tasks and background. We leave out modeling users’ individual traits, as they have trivial impact on our user profiling and are concerned with implementation details of an adaptive system. Table 9 shows the features we intend to capture in our user profiles and their characteristics across the two dimensions discussed above.

5.3.1 A Schema for User Knowledge Representation

Our agent’s value-added services, like enabling users to discover new knowledge (Requirement #6), or aiding them in learning a new topic that they have not seen before (Requirement #4), requires a user profile design that can embed various heterogeneous features, like, knowledge and interests, as we previously described in Section 3.4.

The unit of knowledge representation for a user in our model is a *competency*. According to the definition in [HXC04], a competency is “a *specific, identifiable, definable, and measurable knowledge, skill, ability [...] which a human resource may possess and which is necessary for, or material to, the performance of an activity within a specific business context.*” Draganidis and Mentzas [DM06] further analyzed the term *competency* and outlined four dimensions a competence can be described along: *category* (generic term for a group of similar skills), *competency* (the description of the competence term), *definition* (user scenarios that illustrate this competence) and *demonstrated behaviour* (explanations that clarify if the desired competency has been achieved).

The terms *competence* and *competency* are often used synonymously. However, [Teo06] argues that there is a subtle difference in their meaning: While *competency* is mainly focused on the description of skills a person is supposed to possess in order to achieve a certain target, the term *competence* actually points to the measurement of skills to determine a certain level of expertise.

While we acknowledge this distinction, we consider the terms as synonymous in this dissertation and for the sake of consistency use the term *competence* hereafter.

In our model, users and their competence topics are inter-connected through *competence records*. A competence record contains the provenance metadata of a user’s competences (e.g., the document identifier in which it was found) and can be additionally associated with a *level* of expertise. Since RDF documents intrinsically represent labeled, directed graphs, the semantic profiles of scholars extracted from the documents can be merged through common competence URIs – in other words, authors extracted from otherwise disparate documents can be semantically related using their competence topics.

5.3.2 A Schema for Scholarly User Profiles

Our primary source of competence detection are a user’s publication history, however, a similar approach can be used to bootstrap user profiles for users with no publications, like junior researchers, perhaps by looking into their reading history and collect their ‘interests’.

The idea here is to find the topics mentioned within a user’s publication and use them to represent the user’s background knowledge. The assumption here is that if an author has written about a topic, she must be competent in that topic to various degrees. We use the same semantic modeling practice as explained in Section 5.2 to analyze a user’s publications for rhetorical and named entities.

Each detected named entity can now represent a competence for the use. We retain its corresponding URI on the LOD cloud, the surface form of the topic as written by the author, as well as its start and end offsets as provenance data and create a competence record to store in the user’s profile.

Table 10 shows the vocabularies used to model our semantic scholar profiles and their respective selected terms. We largely reuse IntelLEO⁵ ontologies, in particular, we use the vocabularies from *User and Team Modeling*⁶ and *Competence Management*⁷ ontologies, which are specifically designed for semantic modeling of learning contexts.

Figure 14 shows an example semantic profile in form of an RDF graph. All users in our model are instances of the `<um:User>` class in the User Model (UM) ontology that provides the vocabulary for modeling users in collaborative, learning contexts. As a subclass of the `<foaf:Person>` class, all user instances inherit attributes and relations from the FOAF vocabulary. Therefore, users can be interlinked with external resources on the Web, e.g., finding the same author in a different knowledge base online.

⁵IntelLEO, <http://www.intelleo.eu/>

⁶User and Team Modeling Ontology, <http://intelleo.eu/ontologies/user-model/spec>

⁷Competence Management Ontology, <http://www.intelleo.eu/ontologies/competences/spec>

Table 10: Selected linked open vocabularies for semantic scholar profiles. The vocabulary namespaces used in the table can be dereferenced using the URIs shown in Appendix C.

Domain Concept	Description	LOV Term
User	Scholarly users, who are the documents' authors.	um:User
Competency	Extracted topics (LOD resources) from documents.	c:Competency
CompetencyRecord	A container for provenance metadata of a competence.	c:CompetenceRecord

Property	Type	Domain	Range	LOV Term
hasCompetencyRecord	object	User	CompetencyRecord	um:hasCompetencyRecord
competenceFor	object	CompetencyRecord	Competence	c:competenceFor

5.4 Semantic Modeling of Personal Research Agents

We finally introduce our model for a semantic description of the workflow between a scholar and his personal research agent. While document models and user profiles in the knowledge base are populated as the user interacts with his agent, the metadata and services of the agent are mostly

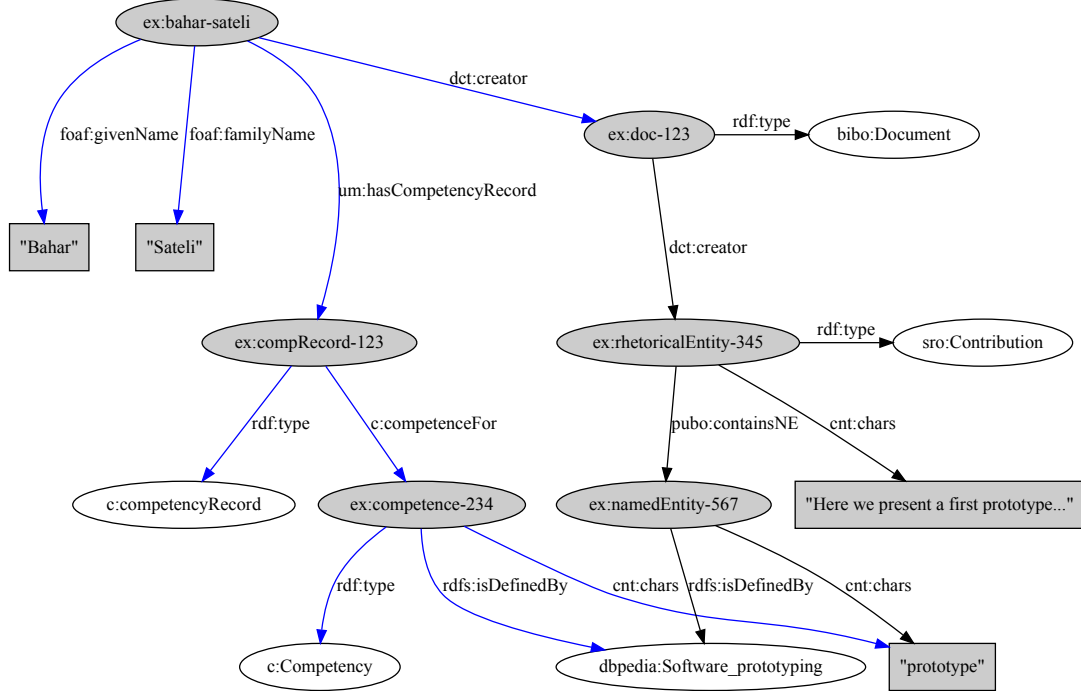


Figure 14: An RDF graph representing a semantic user profile

Table 11: Selected linked open vocabularies for our agent’s workflow. The vocabulary namespaces used in the table can be dereferenced using the URIs shown in Appendix C.

Domain Concept	Description	LOV Term
Agent	A scholarly personal research agent.	prav:PersonalResearchAgent
Artifact	A scholarly artifact, such as a document or dataset that the agent can analyze.	prav:Artifact
Action	An executable operation that the agent can use for semantic computations.	lifecycle:Action
Task	An agent’s unit of work.	lifecycle:Task
TaskGroup	An aggregated collection of work units.	lifecycle:TaskGroup

Property	Type	Domain	Range	LOV Term
performsTask	object	prav:PersonalResearchAgent	lifecycle:TaskGroup	prav:performsTask
task	object	lifecycle:TaskGroup	lifecycle:task	lifecycle:task
resource	object	lifecycle:Action	prav:Artifact	lifecycle:resource
next	object	lifecycle:Task	lifecycle:Task	lifecycle:next
interactsWith	object	um:User	prav:PersonalResearchAgent	prav:interactsWith
interestedIn	object	um:User	prav:Artifact	prav:interestedIn

modelled up-front and may be extended throughout the agent’s lifecycle. A formal semantic description of tasks facilitates consistent implementation of the agent’s services and provides for composing new services by combining various tasks that an agent can perform.

Our Personal Research Agent Vocabulary (PRAV)⁸ is an adaptation of the Lifecycle Schema,⁹ which was originally designed to model the lifecycle of any resource throughout a transition.

An agent’s work unit is a ‘Task’ assigned to it by a user. Tasks are aggregated into ‘Task Groups’ and can be composed in an ordered sequence. While tasks are essentially conceptual entities with properties, such as a description or status, the underlying computations are instances of the ‘Action’ class. Whereas tasks are designed by the agent developers for a specific goal, actions are generic operations, like querying the knowledge base or crawling a repository. In the process, actions can consume, produce or modify ‘Artifacts’. In this dissertation, we restrict our agent’s design to analyze scholarly literature (e.g., journal articles or conference proceedings) as artifacts. Figure 15 shows our agent’s task schema, as well as example instances. For example, a literature review task group shown in the model is divided between two consequent tasks: (i) finding all rhetorical entities

⁸Personal Research Agent Vocabulary, <http://lod.semanticsoftware.info/prav/prav#>

⁹Lifecycle Schema, <http://vocab.org/lifecycle/schema#>

Chapter 6

Automatic Knowledge Base Construction

So far, we established several semantic models for scholarly entities, such as documents and users, which are important to store and manage in the agent’s knowledge base to fulfill its tasks. However, the sheer amount of available and conceivably relevant literature for each user’s tasks and interests makes the manual construction and maintenance of the agent’s knowledge base an unattainable goal. In this chapter, we describe a set of automatic techniques that can populate the agent’s knowledge base with pertinent knowledge, extracted from various artifacts: Bibliographical metadata (Section 6.1), rhetorical and named entities (Section 6.2), as well as user competence records and topics (Section 6.3). We then discuss the transformation of entities to semantic triples (Section 6.4).

6.1 Extraction of Bibliographical Metadata

The schematic structure of scholarly literature often follows a set of conventions established within a discipline’s community (see Chapter 3). However, despite the superficial differences in documents’ formatting, which are proprietary to each publishing venue (e.g., a journal) or company (e.g., Springer), most document schemas follow a set of common patterns. One recurring pattern is dividing the document structure into separate high-level segments, namely, the *front matter* (e.g., article’s authorship metadata), the *body matter* and the *back matter* (e.g., references, appendices). Based on this, automatic extraction of various entities, such as the bibliographical metadata, can be optimized by only examining relevant document segments. For example, the extraction of author names should be focused within the front matter, as several person names might be mentioned in the document’s body matter. Especially when authors are citing other relevant works, there is no easy

way to automatically recognize which of the detected names are the actual authors of the document under study. Therefore, we begin this section by describing an automated method to decompose the full-text of a document for further analysis.

6.1.1 Pre-processing Phase

One of the challenges of automatic processing of text is converting the data from an unstructured or semi-structured format into a homologous representation, suitable for a variety of text mining techniques. A primary step to eliminate the journal- or publisher-specific typesetting of scholarly literature is to strip the documents off of any aesthetic formatting, for example, by scraping the plain-text of a PDF or HTML file. Once the plain-text is available, a number of so-called *pre-processing* [Min12] steps have to be taken to prepare the unstructured text for further semantic analyses. Text pre-processing techniques are language-dependent but common for most text mining tasks.

In our approach, we conduct the following pre-processing steps on all documents prior to any entity detection:

Tokenization, which is the breakdown of unstructured text into smaller, meaningful, discrete lexical units called *tokens*. We classify the tokens into one of *words*, *numbers* or *symbol* categories.

Stemming, which is a normalization process to remove pluralization and other suffixes from word tokens, so that they share homographs.

Sentence Splitting, which is the task of dividing a text into sequences of sentences. Sentences are separated from each other based on the language under study. For example, in English most sentences are divided by a period symbol. Apart from grammatically-correct sentences within an article’s main matter, content in the metadata segments, figures and tables content and captions or code listings are also divided into ‘sentence’ units.

Part-of-Speech Tagging, which is the process of disambiguating the word-category of each word token in a text. A Part-of-Speech (POS) is a category for words that share similar grammatical properties in a sentence, e.g., nouns, verbs, adjectives, or prepositions. Part-of-Speech tagging helps us to focus tasks, such as topic detection, to the relevant tokens, like nouns in a text.

Chunking, which is a shallow analysis of a sentence structure to construct higher order lexical units by combining elementary tokens, such as coupling a determiner and a noun to make a noun phrase.

The decomposed unstructured text is then passed on to the subsequent analysis phases for entity extraction.

6.1.2 Text Segmentation

Following the pre-processing phase, we begin the semantic analysis of unstructured text by dividing the document into several logical and structural segments. In Section 5.2.1, we discussed that Hyppönen and Paganuzzi [HP10] found statistically frequent structural headings in computer science literature that authors tend to use in their manuscripts. Based on their findings, we curated a list of common headers, as well as common linguistic patterns we found in our exemplary datasets to perform an automated segmentation of documents’ full-text on several granularity levels:

On the first level, we divide the document into three coarse-grained segments: the **FrontMatter**, **MainMatter** and **BackMatter**. The **FrontMatter** is the body of text from the Start-of-Document until the beginning of the main body of the document, which is typically the **Introduction** section. The **FrontMatter** segment usually contains the title of the article, its authorship metadata, an abstract, as well as a set of keywords. The entire textual content from the **FrontMatter** segment until the **References** section of the document (if present) is considered the **MainMatter**, which is the body of content used for discourse analysis of the article. Finally, the remaining content of the document is tagged as the **BackMatter** segment that often contains citation information and relevant appendices. An example segmentation is illustrated in Figure 16.

On the second level, we further analyze the content of each segment to break it down into smaller logical sub-segments. This level is required for a zone-specific analysis of text. For example, author name detection in a document will not go through the **Abstract** content, although it is contained within the **FrontMatter** segment. In this step, we perform the following break down:

The **FrontMatter** is divided into the **Title**, **Authorship**, **Abstract** and **Keywords** sections. We consider the **Abstract** section to be within the boundary of the **FrontMatter**, rather than the **MainMatter**, because unlike individual sections of the **MainMatter**, the **Abstract** can be understood as a standalone entity outside of the document’s main body. The **MainMatter** is divided into smaller sections based on our dictionary of section headers and common linguistic patterns mined from our datasets. Finally, the **BackMatter** is divided into **References** and **Appendix** and the **References** section is divided into individual cited works.

6.1.3 Detection of Authorship Metadata

Two pertinent entities for categorical management and retrieval of literature are the author and affiliation names, e.g., in creating an overview of an individual or organization’s contributions (Requirement #5). In this section, we describe a set of pattern-matching rules for detection of such entities. Where possible, we provide a real-world example from our datasets.



Figure 16: Automatic segmentation of a scholarly document

Detecting Author Entities

Person name detection is a well-studied problem in the named entity detection research [NS07]. By convention, author names of a scholarly document are placed in the FrontMatter segment of documents and are composed of several upper-initial tokens in text with optional middle-name initials. We used several online resources and databases of common first names to curate a dictionary that can be matched against the Authorship segment. Matched tokens are then used as clues for annotating authors' full names, for example, using the rules below:

RULE_{author₁}: `DICTIONARYfirstname` + UPPER INITIAL TOKEN + PUNCTUATION + UPPER INITIAL TOKEN

Example (1) "Arne J. Berre"

(<http://ceur-ws.org/Vol-1006/paper5.pdf>)

RULE_{author₂}: `DICTIONARYfirstname` + PREPOSITION + UPPER INITIAL TOKEN

Example (2) "Henk de Man"

(<http://ceur-ws.org/Vol-1006/paper5.pdf>)

RULE_{author₃}: `DICTIONARYfirstname` + UPPER INITIAL TOKEN + HYPHEN + UPPER INITIAL TOKEN

Example (3) “*Hugo Alatrasta-Salas*”

(<http://ceur-ws.org/Vol-1001/paper2.pdf>)

Detecting Affiliation Entities

We designed several rules to automatically capture various patterns of organization names (limited to academic institutions) in documents. The procedure is restricted to the analysis of each document’s **FrontMatter**. We examine the textual content between the last **Author** entity and one of **Email** or **Abstract** segments.

Organization Names. The first step is to compare noun tokens in the **FrontMatter** segment against our dictionary of organizational units. All matched entities will serve as indications of possible mentions of an affiliation. We then progressively apply our affiliation detection rules on sequences of nouns, noun phrases and prepositions in the designated area. The rules below show exemplary patterns and the adjacent URLs are examples from existing publications. The tokens in bold face are terms that match our dictionary.

RULE_{affiliation₁}: UPPER INITIAL NOUN + DICTIONARY_{org} + PREPOSITION + UPPER INITIAL NOUN

Example (4) “*Queensland **University** of Technology*”

(<http://ceur-ws.org/Vol-1518/paper2.pdf>)

Example (5) “*Otto-von-Guericke **University** of Magdeburg*”

(<http://ceur-ws.org/Vol-1315/paper9.pdf>)

RULE_{affiliation₂}: DICTIONARY_{org} + PREPOSITION + UPPER INITIAL NOUN (PHRASE) + CONJUNCTION + UPPER INITIAL NOUN (PHRASE)

Example (6) “*Institute of Knowledge and Language Engineering*”

(<http://ceur-ws.org/Vol-1315/paper9.pdf>)

If a match is found, the start and end offsets of the textual content of the mention is annotated as the begin and end of an **Affiliation** entity. After the first pass through the **FrontMatter**, we then try to find the longest matching pattern by combining intra-organizational units (e.g., research centres, departments and research group naming patterns) and the organization name. If an intra-organizational unit is found in a text in adjacency of an institution name (e.g., a university name), then the longest span of text covering both entities will be annotated as the **Affiliation** annotation. For instance, Example (7) is considered as one affiliation entity, rather than two entities (i.e., one for the school and one for the university). This is because the intra-organizational unit (i.e., the

school) cannot be uniquely identified without its context (i.e., the university), since several schools of dental medicine exist in the world.

RULE_{affiliation₃}: ORGANIZATIONAL UNIT + PUNCTUATION + ORGANIZATION

Example (7) “*School of Dental Medicine, University at Buffalo*”

(<http://ceur-ws.org/Vol-1309/paper3.pdf>)

As a final revision step, we remove any **Author** entities within **FrontMatter** that fall within the boundary of an affiliation name, for example, ‘*Johannes Kepler*’ in Example (8) is part of the university name, rather than an author:

Example (8) “*Johannes Kepler University (JKU)*”

(<http://ceur-ws.org/Vol-1514/paper2.pdf>)

Geographical Locations. We further analyze the annotated **Affiliation** mentions to determine where the organization is located, in particular their country information. In a majority of cases, the location can be identified from the tokens adjacent to the **Affiliation** annotation:

RULE_{affiliation₄}: AFFILIATION + PUNCTUATION + LOCATION

Example (9) “*Otto-von-Guericke University of Magdeburg, Germany*”

(<http://ceur-ws.org/Vol-1315/paper9.pdf>)

There are two ways to tackle this problem: One approach is to keep a dictionary of all the country and city names in the world, which is feasible considering that the list is finite and would hardly change over time. A more reliable solution, however, would be to dynamically find the country names using a Named Entity Recognition (NER) tool. In this approach, the affiliation name and its adjacent noun and noun phrase tokens would go through a disambiguation process and every mention of a country will be tagged as a **Location** entity.¹

Finally, we have to assign a location entity to each affiliation mention. While investigating our dataset documents, we found out that detecting such a relation is a non-deterministic, complex task: First, we observed that the line-by-line scraping of the documents’ text often mixes up the order of the **FrontMatter** tokens, especially in multi-columns formats like ACM. In such cases, the university names are scraped in one line next to each other and the next line has both of their country information. Second, there exists a number of documents in our datasets, which do not have the country name in the document at all. To countervail the disarrayed or missing location information, we can *infer* their relations using a set of heuristics, listed in Algorithm 1:

¹In practice, both approaches can be combined for a polling mechanism, where false positives from the NER process can be cross-validated against the in-house dictionary.

- In the case that there is only one affiliation and one country entity in the metadata body, we can match the two entities together with a high confidence.
- If there is more than one annotation of each type (i.e., **Affiliation** and **Location**), we construct two lists from the annotations sorted by their start offsets in text. We then iterate through the two lists and match each affiliation with a location that has a greater start offset (i.e., mentioned *after* the affiliation), but is located in the shortest distance from the affiliation annotation (in terms of their start offset delta).
- Finally, if no **Location** annotation is available in the **FrontMatter**, we resort to using the DBpedia Lookup² service to run the affiliation name against its ontology. If the affiliation is matched with a resource, we execute a federated query against the public DBpedia SPARQL endpoint,³ looking for triples where the subject matches the affiliation URI and the predicate is one of `<dbpedia:country>` or `<dbpedia:state>` properties from the DBpedia ontology. If such a triple is found, we retain the English label of the object (country) and infer a relation between the affiliation and the country name.

Inferring Author-Affiliation Relations

The last task to be automated in bibliographical entity detection is to associate **Author** entities in the document with their corresponding affiliations. There are three types of relations that we can automatically model:

One-to-One relation, where an author is associated with one affiliation;

One-to-Many relation, where an author is associated with more than one affiliation; and

Many-to-Many relation, where an author is associated with more than one affiliation, and many authors are associated with one affiliation.

An inspection of our dataset documents showed that in the case of many-to-many relations, symbolic clues are integrated within the **FrontMatter** content to associate authors and affiliation entities. Typically, natural numbers or symbols like an asterisk or cross are placed next to the end of each author name, with a matching symbol at the beginning of an affiliation entity. The number or symbol plays the role of an index to associate the two entities for humans comprehension. If such indexes are present in the document, then the agent can automatically associate the entities with high confidence, otherwise a set of heuristics will try to conjecture the relations based on the offset proximity of the entities in text, as listed in Algorithm 2.

²DBpedia Lookup, <https://github.com/dbpedia/lookup>

³DBpedia SPARQL endpoint, <http://dbpedia.org/sparql>

Algorithm 1 Heuristics for inferring affiliation-location relations

Require: A list of organizations O , a list of locations L

Ensure: Set one of $l_j \in L$ as the location for the corresponding $o_i \in O$, *null* otherwise

```
1: procedure INFERENCELOCATION( $O, L$ )
2:   if (size of  $O = 1$  & size of  $L = 1$ ) then
3:     location of  $o_0 \leftarrow l_0$  ▷ clear case of one-to-one mapping
4:   else if (size of  $L = 0$ ) then ▷ look for organization in external ontologies
5:     for all  $o_i$  in  $O$  do
6:        $tup_o \leftarrow$  the triple returned from a DBpedia lookup for  $o_i$ 
7:       if ( $tup_o = null$ ) then ▷ no such resource in the ontology
8:         location of  $o_i \leftarrow null$ 
9:       else
10:         $sub_o \leftarrow$  subject of  $tup_o$ 
11:         $obj_o \leftarrow$  object of a triple where subject is  $sub_o$  and predicate is <dbpedia:country>
12:        if ( $obj_o = null$ ) then ▷ resource has no country information
13:          location of  $o_i \leftarrow null$ 
14:        else
15:           $country \leftarrow$  English label of  $obj_o$ 
16:          location of  $o_i \leftarrow country$ 
17:   else ▷ multiple organization and location entities in document
18:     sort  $O$  by start offset of each  $o_i$ 
19:     sort  $L$  by start offset of each  $l_j$ 
20:     for all  $o_i$  in  $O$  do
21:       for all  $l_j$  in  $L$  do
22:          $s_j \leftarrow$  start offset of  $l_j$ 
23:          $s_i \leftarrow$  start offset of  $o_i$ 
24:         if ( $s_j > s_i$ ) then
25:           location of  $o_i \leftarrow l_j$ 
26:           break;
27:         else
28:           remove  $l_j$  from  $L$ 
```

6.1.4 Detection of References

Surprisingly, detection of references' titles, authors and publishing venues is one of the most challenging parts of document analysis, mostly due to inconsistencies in bibliographical styles used in our datasets. Although bibliography entry formatting is facilitated using authoring tools like BibTeX,⁴ missing or incomplete information are often overlooked in the production or publication stages of an article. Therefore, automatic processing of references in a document requires approaches that are tolerant to incomplete data.

We tackle this problem by hand-crafting rules for multiple bibliography styles, including **abbrv** and **plain** classes⁵ used in our datasets. First, we break down the **BackMatter** segment into smaller fragments: Similar to author names described above, we detect author names and paper title from

⁴BibTeX, <http://www.bibtex.org/>

⁵Bibliography Styles, https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#Bibliography_styles

Algorithm 2 Heuristics for inferring author-affiliation relations

Require: A list of authors A , a list of affiliations O , a flag f for whether indexes exist in the document

Ensure: Set one or more of $o_j \in O$ as the association for the corresponding $a_i \in A$, *null* otherwise

```
1: procedure INFERRASSOCIATION( $A, O$ )
2:   if ( $f = \text{true}$ ) then
3:     for all  $a_i$  in  $A$  do
4:        $index_a =$  index symbol of  $a_i$ 
5:       for all  $o_j$  in  $O$  do
6:          $index_o =$  index symbol of  $o_j$ 
7:         if ( $index_a = index_o$ ) then
8:           associate  $a_i$  with  $o_j$ 
9:   else
10:    sort  $A$  by start offset of each  $a_i$ 
11:    sort  $O$  by start offset of each  $o_j$ 
12:    for all  $o_i$  in  $O$  do
13:      for all  $l_j$  in  $L$  do
14:         $s_j \leftarrow$  start offset of  $o_j$ 
15:         $s_i \leftarrow$  start offset of  $a_i$ 
16:        if ( $s_j > s_i$ ) then
17:          associate  $a_i$  with  $o_j$ 
18:          break;
```

each reference. We then annotate the tokens in between the paper title and the year of publication (or End-of-Line character) as the publishing venue. References are eventually categorized into either ‘*journal*’ or ‘*proceedings*’ classes, based on whether a journal citation (volume, number and pagination) is present, like the ones shown below. The tokens in bold face are terms that match our dictionary:

RULE_{reference₁}: (AUTHOR + PUNCTUATION)* + (TOKEN)* + (DICTIONARY_{j_{rnl}})* + PREPOSITION+ NOUN PHRASE + PAGINATION + YEAR

Example (10) “G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru, and S. Decker. *Sig.ma: Live views on the web of data*. **Journal** of Web Semantics, 8(4):355-364, 2010”

(<http://ceur-ws.org/Vol-813/ldow2011-paper10.pdf>)

Example (11) “M. Hausenblas, “Exploiting linked data to build applications,” *IEEE Internet Computing*, **vol. 13**, **no. 4**, **pp.** 68-73, 2009”

(<http://ceur-ws.org/Vol-813/ldow2011-paper12.pdf>)

6.2 Extraction of Rhetorical and Named Entities

In Section 5.2.2, we explained how we semantically model the scientific discourse of a document using its rhetorical and named entities. Here, we describe how we can automatically capture REs and NEs using a set of custom patterns that we gathered from an extensive study of the domain literature.

6.2.1 Common Linguistic Patterns in Rhetorical Entities

In our design, we model a document’s rhetorical entities on a sentential level, that is, we categorize each sentence of a document as whether it represents a rhetorical move or not. We curated grammatical structures and discourse markers that would indicate a possible presence of a rhetorical entity in a given document and further aid in its classification. In our methodology, detection and classification of rhetorical entities is performed in an incremental fashion:

The first step is to find *metadiscourse* elements in a document. Metadiscourse are those aspects of text that present the authors’ stance towards the readers [Hy198]. In particular, we are interested in finding textual metadiscourse elements that function as *frame markers* [Hy198] in a document. Frame markers are explicit references to discourse acts within a text, such as preparing the readers for an argumentation. In scholarly literature, authors use metadiscourse elements to organize their discourse goals and “*establish preferred interpretations of propositional meanings*” [Hy198]. Introductory sentences in the **Abstract** section of an article are such metadiscourse elements that authors use to convey an overview of their work described in the document.

Based on our observations, metadiscourse entities often contain a discourse *deixis*. Deictic phrases are expressions within an utterance that refer to parts of the discourse and cannot be understood by readers without contextual information. For example, the word “*here*” in “*here, we describe a new methodology...*” refers to the article that the user is reading. We devised a set of common patterns, extracted from study of computer science articles from various disciplines, to identify variations of deictic phrases and subsequently, detecting metadiscourse elements. The patterns are complemented with a manually-curated dictionary of recurrent nouns and noun phrases in metadiscourse elements.

The following are patterns based on pre-defined sequences of grammatical tokens and entries from our dictionary (deictic phrases are in bold), followed by real examples from the literature:

RULE_{deictic₁}: DETERMINER + NOUN PHRASE_{dictionary}

Example (12) “***This paper*** presents a use case of adding value to a bird observation dataset...”

(<http://ceur-ws.org/Vol-1155/paper02.pdf>)

RULE_{deictic₂}: PREPOSITION + DETERMINER + NOUN PHRASE_{dictionary}

Example (13) “***Throughout this paper***, a total of six video steganography tools have been...”

(<http://peerj.com/articles/cs-7>)

Example (14) “***In this manuscript***, I report a Python command-line tool, *ngg2*, for ...”

(<http://peerj.com/articles/cs-33>)

RULE_{deictic₃}: ADVERB + PRONOUN_(I|We)

Example (15) “**Here**, **we** demonstrate how our interpretation of NPs, named graphs, knowledge...”

(<http://ceur-ws.org/Vol-721/paper02.pdf>)

Based on the detected deictic phrases, we capture metadiscourse phrases in a sentence using verbs and phrases from our dictionary of rhetorical verbs:

RULE_{metadiscourse₁}: DEICTIC PHRASE + VERB_{presentation}

Example (16) “**This paper presents** a use case of adding value to a bird observation dataset...”

(<http://ceur-ws.org/Vol-1155/paper02.pdf>)

RULE_{metadiscourse₂}: DEICTIC PHRASE + PRONOUN + VERB_{presentation}

Example (17) “**Here**, **we demonstrate** how our interpretation of NPs, named graphs, ...”

(<http://ceur-ws.org/Vol-721/paper02.pdf>)

The complete list of our rules for marking deictic and metadiscourse phrases in a text is provided in Appendix E.

Once the boundaries of all rhetorical entities are annotated in a text, we can further classify them into a *rhetorical move*. In this dissertation, we focus on classifying the REs into either Contributions or Claims classes, which are sufficient for the agent’s tasks (Requirements #1, #3 and #5).

Contributions. We designed hand-crafted rules to recognize Contribution (see Definition 5.2.2) sentences by detecting grammatical structures often observed in scientific argumentation to describe the authors’ contributions. The rules look at sequences of deictic phrases, metadiscourse mentions, the rhetorical functions of the verbs mentioned in the sentence and the adjacent noun phrases to classify a sentence as a Contribution, as in the following example (matching string is in bold):

RULE_{contribution₁}: METADISCOURSE + NOUN PHRASE

Example (18) “**This paper presents a use case** of adding value to a bird observation dataset...”

(<http://ceur-ws.org/Vol-1155/paper02.pdf>)

RULE_{contribution₂}: METADISCOURSE + ADVERB + NOUN PHRASE

Example (19) “**Here**, **we demonstrate** how our interpretation of NPs, named graphs, knowledge...”

(<http://ceur-ws.org/Vol-721/paper02.pdf>)

Claims. The extraction of Claim entities (see Definition 5.2.2) is performed similar to the Contribution annotations and conducted based on deictic phrases detected in a text. However, here we require that the deictic phrases in Claim sentences explicitly refer to the authors’ contributions presented in the paper. Hence, we distinguish Claims from other classes in the way that the sentence containing the deictic phrase must (i) be a statement in form of a factual implication, and (ii) have a comparative voice or asserts a property of the author’s contribution, like novelty or performance:

RULE_{claim₁}: METADISOURSE + DETERMINER + ADJECTIVE + (TOKEN)* + DOMAIN CONCEPT

Example (20) “*We built the first BauDenkMalNetz prototype...*”

(<http://ceur-ws.org/Vol-721/paper04.pdf>)

RULE_{claim₂}: DEICTIC PHRASE + VERB + DOMAIN CONCEPT

Example (21) “*Our approach is compatible with the principles...*”

(<http://ceur-ws.org/Vol-903/paper02.pdf>)

6.2.2 Detection of Domain Concepts as Named Entities

We focus on the automatic extraction of domain concepts in the MainMatter and the Abstract and Title entity in the FrontMatter segments. We annotate every mention of a domain concept, such as tools, frameworks, techniques, algorithms, methods and datasets in the text. However, since there can be potentially thousands of domain concepts in each document, we can automate the detection by using a Named Entity Recognition (NER) tool, like the DBpedia Spotlight service. Ideally, if the NER tool is also capable of linking the mention to a resource in an ontology (referred to as *grounding* the entity), this approach has the advantage that all the domain concepts have machine-readable information attached to them.

6.3 Scholarly User Profile Population

Asking users to populate the scholarly profiles modelled in Section 5.3 with potentially hundreds of topics they are competent in is impractical. Therefore, we aim at automatically *bootstrapping* the user profiles with any available content associated with a user, e.g., a set of documents the user has (co)-authored. We devised an automatic workflow for the construction of scholarly profiles that largely reuses techniques we described earlier in document analysis, based on the assumption that topics mentioned in a document present a collective set of competences of its authors.

The input to the workflow is a set of scholarly documents. Since it is not feasible to manually construct and maintain a knowledge base of all possible competence topics, again, we leverage

the LOD cloud as a source of continually-updated knowledge. Our idea is to use an NER tool to ground each named entity within a user’s publications to a URI (see Section 5.2.2). Further grammatical processing steps are performed to filter out named entities that do not typically represent competences, like adverbs or pronouns. We exclude processing the sentences in figure and table captions, formulas, section headers and references, as we empirically verified that these document regions rarely contain the authors’ competence topics. Finally, the user profile is populated with the detected topics. Each topic is wrapped up in a competence record annotation that retains where in a document the competence is found. We use the raw frequency of the detected topics (named entities) in documents as a means of ranking the top competence topics for each scholar.

Based on our user profile design, our automatic approach creates a unique competency record for each detected topic in a user’s publication. Since all competency records specify a topic (as a named entity grounded to a URI on the LOD cloud), we can determine a user’s background knowledge using two criteria: (i) the distinct set of competences (i.e., all competence triples dereferencing the same URI) represents the collective set of topics a user knows about, and (ii) for each known topic, the total number of its related competence records in a profile is an indicator of the user’s competence level.

An important choice in our automatic user profiling is deciding whether all topics in a document are indeed representative of its authors’ competences. Or perhaps, a subset of the topics located in the *rhetorical zones* of an article are better candidates? To test this hypothesis, we keep an additional feature for each competence topic that indicates it was mentioned within the boundary of a rhetorical zone in the document (e.g., a topic within a Contribution sentence). We will revisit our hypothesis in Section 8.3.

6.4 Triplification: Transforming Annotations to Triples

So far, we explained how we can automate the process of semantic modelling of the agent’s working context (i.e., documents and users). We also established that all the extracted information are captured in form of annotations. Depending on what text mining framework is used to implement the information extraction process, the resulting annotations will have some proprietary format. One of our contributions in this dissertation is a novel technique for converting the results of an NLP pipeline into an LOD-compliant knowledge base for the agent. Representing the NLP results using W3C standards [CWL14], such as the Resource Description Framework (RDF) and RDF Schema (RDFS), provides high flexibility, as the same NLP pipeline can drive the generation of triples in different knowledge bases with different vocabularies. It also relieves the NLP engineer from dealing with the technical details of generating correct Uniform Resource Identifiers (URIs),

thereby providing an agile solution for bringing NLP results onto the Linked Open Data (LOD) cloud.

We propose a high-level vision for the design of a component that can be added to an NLP pipeline, thereby providing functionality to export the text analysis results in linked data format. We derived a number of detailed requirements from this vision:

Scalability. For the population of large knowledge bases, it must be possible to scale out the generation of triples, i.e., running pipelines on multiple (cloud) instances and storing the resulting triples in a networked triplestore. The time required to export the annotations as triples must scale linearly with the size of the documents and the number of triples to be exported (see Requirement #10).

Separation of Concerns. Apart from adding a new component to an analysis pipeline, no further changes must be required on the NLP side. Thus, we separate the work of a language engineer, developing the NLP pipeline (e.g., to find domain-specific entities) from the work of a knowledge base engineer, who defines the structure of a concrete knowledge base.

Configurability. The export must be dynamically configurable, so that the same NLP pipeline can drive the generation of different triples in the same, or multiple different, knowledge bases. In particular, the vocabularies used in the mapping process must be easily changeable, to support experiments with different knowledge bases and different application scenarios.

LOD Best Practices. The solution must conform to the relevant W3C standards on Linked (Open) Data. This includes the recommended format for the generated triples, as well as the support of standard protocols for communicating with triplestores in a product-independent fashion.

6.4.1 Mapping Language

The central idea of our approach to automate knowledge base construction from NLP is to *externalize* the knowledge about the export process, so that the NLP pipeline can remain mostly unchanged. This provides for the required separation of concerns and makes it possible to easily reuse existing NLP pipelines for different knowledge bases. Hence, the exact same pipeline can be used to generate different LOD triples, e.g., when facts need to be expressed with different LOD vocabularies for different applications.

In our approach, how the NLP results are mapped to triples is defined using a declarative language. This *mapping* language provides constructs for mapping entities to triples, specifically:

Entities, which are annotations of interest to be exported as subjects;

Features, which are features of an annotation to be exported as properties and objects related to the given subject (entity); and

Meta-Information, which are information about the export process, such as text offsets of entities or the name of the pipeline that detected an entity.

A concrete mapping *configuration* is a set of rules (see Figure 17 for an example). These rules are read by the export component, typically at the end of a pipeline, and triples are then generated according to these rules. The configuration can be read from the same knowledge base, which provides for a self-contained NLP export.

Mapping Entities

For each entity that needs to be exported, a corresponding mapping rule declares the NLP type and its output RDF type. For example, an **Author** entity detected in the front matter of a document can be mapped using the FOAF vocabulary. Lines 9–11 of Figure 17 show an example mapping rule.⁶ The rule reads as follows:

There is a mapping identified by `<ex:AuthorEntity>` that exports any annotation of type ‘Author’ to an RDF subject triple. The URI generated for this new triple must use ‘<http://semanticsoftware.info/author/>’ as its base. All subject triples have a `<rdf:type>` predicate, connecting the subject to the `<foaf:Person>` class in the FOAF ontology.

This mapping approach makes our approach extremely flexible: For instance, in order to change the generation of triples to use the **Person Core Vocabulary**,⁷ instead of FOAF, only a change of the mapping rule in the configuration file is required, which is read at export-time. Thus, the designer of a knowledge base is free to experiment with multiple different knowledge representation approaches, without requiring any reconfiguration on the NLP side.

Mapping Features

Most entities are further described in an NLP process with additional features, e.g., the gender of an author. We can map any existing feature of an entity (one subject URI) to different properties and objects, with the same configuration approach. For example, to map a feature ‘*gender*’ for an **Author**, we would define an additional mapping rule to represent it using a `<foaf:gender>` property in the FOAF ontology, as shown in lines 23–25. To declare a feature mapping for entity export, we

⁶The prefix `map:` refers to our mapping language. See Appendix C for more details.

⁷Person Core Vocabulary, <https://www.w3.org/ns/person>

```

1 @prefix map: <http://semanticsoftware.info/mapping/mapping#> .
2 @prefix rel: <http://purl.org/vocab/relationship/> .
3 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
4 @prefix cnt: <http://www.w3.org/2011/content#> .
5 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
6 @prefix ex: <http://example.com/> .
7
8 ### Annotation Mapping ###
9 ex:AuthorEntity a map:Mapping ;
10     map:type      foaf:Person ;
11     map:GAType    "Author" ;
12     map:baseURI   "http://semanticsoftware.info/author/";
13     map:hasMapping ex:GenderMapping ;
14     map:hasMapping ex:ContentMapping .
15
16 ex:AffiliationEntity a map:Mapping ;
17     map:type      foaf:Organization ;
18     map:GAType    "Affiliation" ;
19     map:baseURI   "http://semanticsoftware.info/affiliation/";
20     map:hasMapping ex:ContentMapping .
21
22 ### Feature Mapping ###
23 ex:GenderMapping a map:Mapping ;
24     map:type      foaf:gender ;
25     map:feature    "gender" .
26
27 ex:ContentMapping a map:Mapping ;
28     map:type      cnt:chars ;
29     map:feature    "content" .
30
31 ### Relation Mapping ###
32 ex:AuthorAffiliationRelationMapping a map:Mapping ;
33     map:type      rel:employedBy ;
34     map:domain    ex:AuthorEntity ;
35     map:range      ex:AffiliationEntity ;
36     map:feature    "employedBy" .

```

Figure 17: Example rules, expressed in RDF, declaring how NLP annotations should be mapped to semantic triples for automatic knowledge base population

only need to create a new predicate in the configuration file, adding `<ex:GenderMapping>` to the `<ex:AuthorEntity>`, as shown in line 13.

Additionally, we provide for the export of several meta-information of an annotation, such as the start and end offsets of the entity in a document, as well as the underlying string representation (surface form), as shown in lines 27–29 of our example configuration file.

Mapping Relations

Finally, we can export arbitrary relations between entities into RDF triples. There are two kinds of relations that can be described using our mapping language:

Entity Relations are user-defined associations between entities that annotations represent. For example, an ‘*employedBy*’ relation between an author and an affiliation in the front matter of a document can be defined with a relation, where the domain is a specific **Author** annotation and the range can be one or more **Affiliation** annotations in the document. In the export process, an additional triple will be generated with the two subject triples representing the annotations and the user-defined semantic type as the predicate.

Annotation Relations concern the association between annotations as they were found in a document. For example, a ‘*contains*’ relation describes if one annotation falls within the boundary of another, based on their text offsets. This kind of relation is beneficial, for example, when we would like to model if a named entity (topic) falls within the rhetorical zones of a document.

Lines 32–36 of Figure 17 show an entity relation between authors and affiliations in scholarly literature, as explained above. Additionally, we provide for exporting a number of meta-information of an export process as relations, including: the name of the NLP pipeline that generated the annotations, creation timestamp, document name, corpus name, among others.

6.4.2 URI Generation

An important design feature of the triplification process is the generation of URIs for the resulting triples from text. In designing the URI generation scheme, we had to strike a balance between (i) conforming to LOD best practices; (ii) taking into account the NLP source of the URIs; and (iii) their usability, in particular for querying knowledge bases that mix NLP-generated knowledge with other sources.

Figure 18 shows an example for a subject URI that was generated for a single **Author** instance in a document, according to Figure 17’s export rules. Conforming to LOD best practices, URIs are HTTP-resolvable. The triplification component allows the user to specify the *base URI* (e.g., for a public SPARQL endpoint), which should always be a domain ‘owned’ by the user. This is followed by a *unique run id*, which ensures that each new run of the component on the same text generates new URIs. Generally, exposing implementation details in URIs is discouraged [WZRH14]. However, re-running NLP pipelines (and therefore re-generating triples) is an extremely common occurrence in language engineering – for example, after improving a component, a machine learning model, or experimenting with different parameters. Hence, we decided that the triples from different runs of

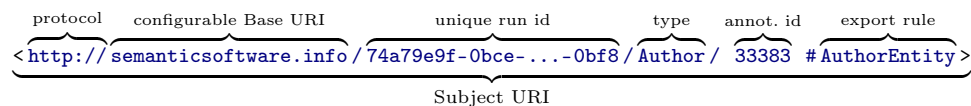


Figure 18: Anatomy of a generated URI for an **Author** annotation

an NLP pipeline must peacefully co-exist in a knowledge base and be easily distinguishable, which is achieved with the generation of this *run id*. Next, the semantic (annotation) *type* as detected by the NLP pipeline is encoded (here **Author**), followed by its unique *annotation id* within a document. Finally, the mapping *export rule* name is added, as it is possible to export the same annotation (with the same type) multiple times, using different export rules (e.g., with different vocabularies).

6.5 Summary

In this chapter, we looked into how we can automate the population of our agent’s knowledge base. We showed how we can automatically annotate pertinent knowledge extracted from documents and user’s relevant publications using text mining techniques. We also showed a robust, flexible methodology to transform the extracted knowledge into semantic triples based on the RDF framework. In the next chapter, we will provide the implementation details of the techniques described here to populate a knowledge base.

Chapter 7

Implementation

In this chapter, we describe the language processing components and the overall workflow that we implemented to reflect the system design illustrated in the preceding chapters. Most of the components, especially the linguistic analysis processing resources, are developed based on the *General Architecture for Text Engineering* (GATE)¹ framework [CMB⁺11]. Please refer to Section 3.1 for the related terminology and foundations of working with GATE.

7.1 Document Pre-processing Pipeline

As we explained earlier in Section 6.1, prior to any semantic analysis of scientific literature, all documents must go through a pre-processing phase, where the syntactical structure of their content is annotated for the downstream processing resources. Most pre-processing tasks involved in this phase are domain-independent and can be reused across several text mining tasks, as long as the natural language of the text (e.g., English) is known. GATE, as a general architecture for developing language processing pipelines, readily offers a variety of pre-processing Processing Resources (PRs) for several languages, including English. We largely reuse the PRs from GATE’s *ANNIE* [CMBT02] and *Tools* plugins. Specifically, we use the following processing resources:

Document Reset PR removes any pre-existing annotations from a document, e.g., from previous runs of the pipeline. This processing resource can be configured to omit certain annotation types or annotation sets when resetting the document, which is convenient when evaluating the performance of our pipelines over gold standard documents with manual annotations.

ANNIE English Tokeniser breaks the stream of a document’s text into tokens, classified as words, numbers or symbols. The output of this processing resource are **Token** annotations

¹GATE, <http://gate.ac.uk>

in the document. Each **Token** annotation holds a feature map of its kind (e.g., word, symbol), character length, the orthographical form (e.g., all caps, upper initial) and its string content (surface form).

RegEx Sentence Splitter is a flexible approach to detect boundaries of sentences in a document. The input to this processing resource is a set of Java-style regular expressions to define what constitutes a split characters in the text and a list of non-split text fragments, such as full-stops within abbreviations. The output of this processing resource are **Sentence** annotations in the document.

ANNIE POS Tagger is a modified version of Mark Hepple’s [Hep00] Brill-style tagger that adds a part-of-speech feature to each **Token** annotation in a text. This processing resource uses a pre-defined default lexicon and a machine learning model trained on a large-scale news corpus for POS tagging. The assigned tags are a superset of the PENN TreeBank [MMS93] POS categories and include 43 classes for word tokens and 11 classes for symbols, like punctuations. Please refer to Appendix B for the description of POS tags used in this manuscript.

GATE Morphological Analyser takes in **Token** annotations and performs lemmatization on their surface forms. The canonical form of each token is added as the **root** feature to its corresponding **Token** annotation. Considering the canonical form of tokens facilitates working with tokens and matching various inflected forms of the same word against our curated dictionaries (e.g., the rhetorical verbs).

MuNPEX English NP Chunker is a processing resource that can detect noun phrase chunks in a sentence and further analyze them for their head noun and any additional pre- or post-modifiers.² The output of this processing resource are **NP** (noun phrase) annotations in a document. Detecting noun phrases facilitates filtering the named entities in a text that correspond to domain topics, as explained in Section 5.2.2.

The above processing resources execute sequentially with a pre-defined order over a document’s full-text content. Each processing resource adds a new annotation type to the document or modifies the feature of an existing one. Figure 19 shows the pre-processing pipeline in the GATE Developer environment. Table 12 shows the annotations generated by each processing resource, executed on an example text.

²Multi-lingual Noun Phrase Extractor (MuNPEX), <http://www.semanticsoftware.info/munpex>

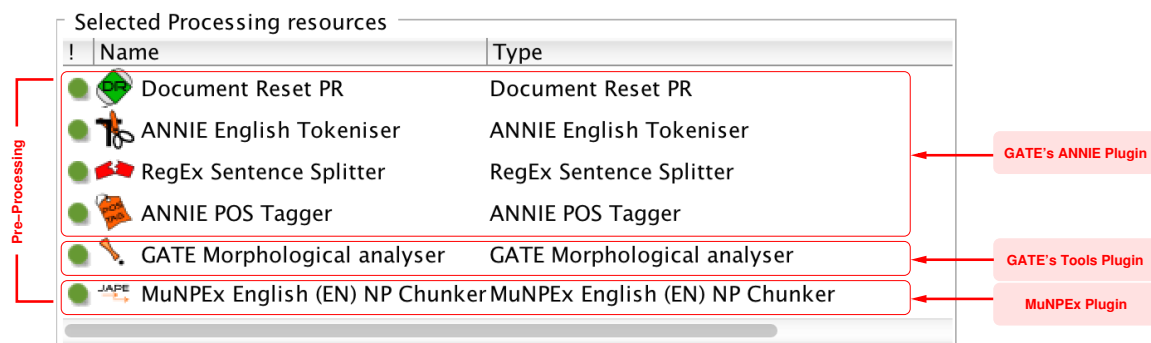


Figure 19: The sequence of processing resources in the pre-processing pipeline

that internally uses Xpdf⁶ to scrape the text of a given scientific article and applies Conditional Random Fields (CRF) models to automatically label segments of a text into one of 55 classes in its schema, including vocabularies for headers, sections, and publishing metadata.

If neither of the above situations applies on the input document, then the ‘*Text Segmentation*’ pipeline attempts to conjecture the structure of a document with our hand-crafted rules, written in form of JAPE grammars (See Section 3.1), combined with a dictionary of common header names in computer science articles (manually constructed from our training data, as well as the list from [HP10]). We first annotate the beginning and end of each document with Start-Of-Document (SOD) and End-Of-Document (EOD) annotations that serve as pivotal points. We then match our gazetteer of section names to find conventional headers like ‘*Introduction*’, ‘*Abstract*’ or ‘*References*’. All other sequences of word tokens, started by the Start-Of-Line (SOL) character or a number, are blindly annotated until we reach the End-Of-Line (EOL) character. Once the section headers are identified, every token (words, number and symbols) in between each detected header and the next one becomes a *Section*, using the detected header as its identifier. Using this approach, on the highest-level we can decompose the full-text of each document into the *front matter*, *main matter* and *back matter* segments, as explained in Section 6.1.

We also annotate the boundaries of floating elements, such as tables, code listings and figures inside documents. They not only help to construct a finer-grained structural model of the document, but we can later omit further semantic analysis of the floating elements, since they are rarely written in a natural language or represent any domain concepts that the agent is interested to model in its knowledge base. We curated a set of *trigger* words that are used in figure and table captions, such as ‘*Fig.*’, ‘*Figure*’ and ‘*Table*’, in a gazetteer. The detection of captions is conducted in two steps: First, the trigger words from our gazetteers are annotated in a text. Then, a JAPE grammar looks for sequences of numbering patterns (using Roman numerals, Arabic numerals and sub-numbering),

⁶Xpdf, <https://www.xpdfreader.com/>

adjacent to the annotated trigger words. The sequences of tokens succeeding the caption number are annotated as the label for the float and annotated with the corresponding type (table, figure or listing) for the element. In case where we cannot find a number in the caption, a counter incrementally numbers the generated annotations, ordered by their starting offset.

Finally, we further analyze the *back matter* segment for the detection of cited publications in a document. In practice, this structural analysis proved to be a rather challenging task, mostly due to inconsistencies in bibliographical styles used by authors. We found numerous examples in our datasets, where authors organized their references manually and polluted the entries with invalid patterns, spelling mistakes and unconventional abbreviations for long conference or journal names.

We tackled this problem by hand-crafting JAPE rules for multiple styles, including **abbrv** and **plain** classes used in the training set, which are fault-tolerant to omissions of symbols prescribed by a certain style. The analysis is limited to the text covered by a **References** annotation, that is, the tokens after the *References* section header, until the EOD or possibly the next section header representing the appendices of the article.

7.2.2 Authorship Metadata Extraction

Once the document segments and their boundaries are extracted from a text, we focus the bibliographical metadata analysis of each document to the scope of its *front matter*. Again, if no machine-readable metadata, such as the ones provided by GROBID or PeerJ Schema, is available in the original markup of the document, our ‘*Authorship Metadata Extraction*’ pipeline tries to find the required semantic entities based on the linguistic characters of tokens within the *front matter* of the document. We implemented the heuristics explained in Section 6.1.3 as multiple processing resources, based on the GATE Embedded libraries.

Author Full Name Detection

The person name detection is based on tokens marked by our gazetteer of common first names, an extended version of the ANNIE gazetteer. We implemented several JAPE grammars that look at word tokens following the marked first names and try to mark up the full name of authors. Using our JAPE rules, we can detect authors’ names with full middle names, abbreviated middle names, and family names with special characters, like hyphens or apostrophes, and annotate them as **Authors** in a text.

Affiliation Name and Unit Detection

In Section 6.1.3, we presented several patterns to capture various forms of organization names, limited to academic institutions. We hand-crafted one or more rule-based grammars corresponding to

each pattern in the JAPE language. The developed grammars capture a wide variety of organizational names and additionally try to find the geographical location of the organization from (i) the name of the institution or (ii) the location name mentioned closest to the organization, in terms of its start offset in a text. We retain the detected location name, along with the Affiliation annotation.

Relation Extraction

Once the Author and Affiliation annotations are extracted from the front matter of the document, we can now start inferring the semantic relations between them. We developed two separate GATE processing resources, ‘*Author-Affiliation Relation Inferrer*’ and ‘*Affiliation-Location Relation Inferrer*’ and implemented the heuristics in Algorithms 1 and 2 in Java, based on the GATE Embedded libraries. These processing resources, respectively, can extrapolate where an Affiliation is located and which Authors are employed by a detected Affiliation entity. The relation extraction processing resources are, however, different from entity detection resources. In entity extraction PRs, like author name detection, the input to the pipeline is the plain text, as well as other annotations, and the output is either a new annotation added to the document or a new feature placed in the feature map of the designated annotations. In relation extraction PRs, the input to the pipeline is a set of annotations and the generated output are *relation* annotations, a particular type of GATE annotations that are not bound to any offsets in text, but the document as a whole. Each relation is unique in the document, holds a *type* and has two or more *member* annotations.

The Author-Affiliation Relation Inferrer PR generates relations of type ‘*employedBy*’ between one Author annotation and one or more Affiliation annotations in a text, as justified in Section 6.1.3. In contrast, the Affiliation-Location Relation Inferrer PR creates a one-to-one relation between one Affiliation and one Location mention in the front matter, using ‘*locatedIn*’ as its semantic type.

7.3 Discourse Analysis Pipeline

We now present our text mining pipeline that can analyze the main matter of each document with the goal of creating a semantic model of its rhetorical structure. The implementation of our discourse analysis pipeline is divided into two separate, stand-alone plugins that we developed for the GATE environment, namely, *Rhetector* for rhetorical entity extraction and *LODtagger*, a GATE wrapper for annotating scientific literature with external named entity recognition tools.

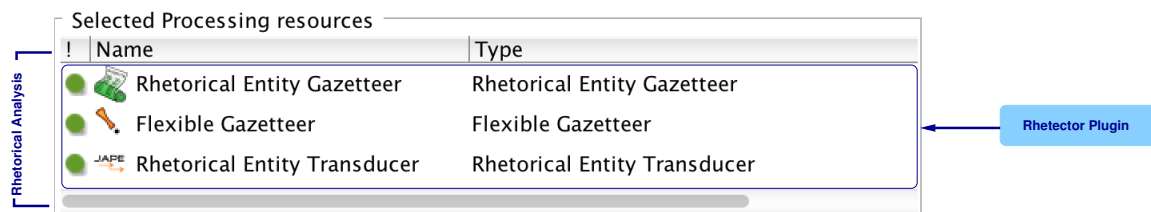


Figure 20: The sequence of processing resources in the Rhetector pipeline

7.3.1 Rhetector: Automatic Detection of Rhetorical Entities

We developed Rhetector⁷ as a stand-alone GATE plugin to extract rhetorical entities from scientific literature. Rhetector has several processing resources, as shown in Figure 20:

Rhetorical Entity Gazetteer produces **Lookup** annotations by comparing the text tokens against its dictionary of domain concepts, deictic phrases and rhetorical verbs. Note that, as we discussed in Section 5.2.2, the gazetteers only include the canonical form of all the interesting tokens, i.e., the singular form for all nouns and the base form for all verbs. In order to match various inflected forms of tokens in a text against the dictionary, the Rhetorical Entity Gazetteer uses the *Flexible Gazetteer* processing resource, which can compare the gazetteer entities against the root form of **Token** annotations.

Rhetorical Entity Transducer applies the rules described in Section 6.2.1 to sequences of **Tokens** and **Lookup** annotations from the gazetteer PR to extract rhetorical entities. The rules are implemented using GATE’s JAPE language (see Section 3.1) in form of regular expressions over document annotations. This processing resource creates incremental annotations, starting with finding Deictic phrases, detecting **Metadiscourse** annotations and finally classifying every sentence with a metadiscourse phrase into one of our rhetorical types.

Figure 21 shows a sequence of JAPE rules for extracting a **Contribution** sentence. Lines 1–6 in Figure 21a show a JAPE rule that matches an upper-initial adverb (e.g., ‘*In*’), followed by a determiner (e.g., ‘*this*’) and a **Lookup** annotation generated by the Rhetorical Entity Gazetteer, and creates a **Deictic** annotation covering the matched span in a text. Similarly, lines 8–13 create a metadiscourse overlapping both a **Deictic** annotation and a rhetorical verb. Lines 15–18 classify the sentence containing the **Metadiscourse** annotation as a **RhetoricalEntity**, copying over the rhetorical types from the metadiscourse annotation within its boundary. Figure 21b shows the generated annotations, colour-coded in GATE’s graphical user interface.

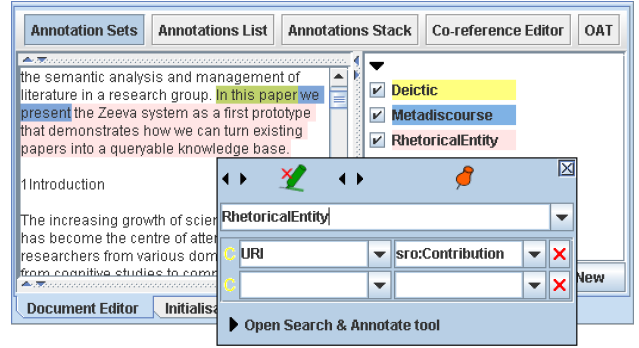
⁷Rhetector, <http://www.semanticsoftware.info/rhetector>

```

1 Rule: INDeictic (
2   {Token.category == "IN", Token.orth == "upperInitial"}
3   {Token.category == "DT"}
4   {Lookup.majorType == "DEICTIC"}
5 ):mention -->
6 :mention.Deictic = {content = :mention@string}
7
8 Rule: ContributionActionTrigger (
9   {Deictic} {Token.category == "PRP"}
10  ({Token.category == "RB"})?
11  {Lookup.majorType == "ACTION"}
12 ):mention -->
13 :mention.Metadiscourse = {type = "sro:Contribution"}
14
15 Rule: RESentence (
16   {Sentence, Sentence.contains ({Metadiscourse}):meta}
17 ):mention -->
18 :mention.RhetoricalEntity = {URI = :meta.type}

```

(a) Example JAPE rules



(b) Detected RE annotation in GATE Developer

Figure 21: JAPE rules (a) to extract a Contribution sentence and the generated annotations in GATE Developer (b)

7.3.2 LODtagger: Named Entity Detection and Grounding

In Section 6.2.2, we mentioned that we take advantage of generic named entity recognition tools to annotate every mention of a domain concept, such as names of tools, algorithms or frameworks in the full-text of scholarly literature. To this end, we implemented an extensible GATE plugin, called *LODtagger*,⁸ that can act as wrapper for any given NER tool. As a concrete application, we locally installed the DBpedia Spotlight⁹ tool [DJHM13] version 0.7¹⁰ and use its RESTful annotation service to find and disambiguate named entities in our documents. The LODtagger pipeline consists of two processing resources, as shown in Figure 22:

DBpediaTagger sends the full-text of documents to Spotlight as an HTTP POST request and receives a JSON array as the result. It then parses each JSON object and adds a **DBpediaLink** annotation, with a DBpedia URI as its feature, to the document.

DBpedia_NE_Filter filters the resulting entities by aligning them with noun phrases (NPs), as detected by the *MuNPEx NP Chunker* for English during pre-processing. The aligning is performed using a JAPE rule that removes **DBpediaLink** annotations which are not nouns or noun phrases. Similarly, we discard NEs that include a pronoun only.

Figure 23 shows an example Spotlight service call on a document and the backend response in form of a JSON array. Each JSON object represents a named entity with its corresponding LOD

⁸LODtagger, <http://www.semanticsoftware.info/loddagger>

⁹DBpedia Spotlight, <http://spotlight.dbpedia.org>

¹⁰with a statistical model for English (en_2+2), <http://spotlight.sztaki.hu/downloads/>

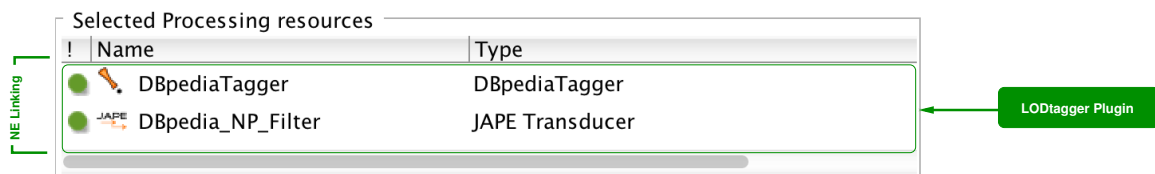


Figure 22: The sequence of processing resources in the LODtagger pipeline

resource stored in the URI key. For each named entity, we have the starting offset of its mention in text (starting from 0 signalling the beginning of the text), with which we can calculate the end offset from the length of its surface form. Each linked named entity also has a similarity score that demonstrates how confident Spotlight was in grounding the named entity to its URI. The LODtagger pipeline allows for customizing the confidence and similarity score threshold of the Spotlight service to remove possible false positive matches, at the cost of decreasing recall.

7.4 ScholarLens: Semantic User Profiling Pipeline

Our *ScholarLens* pipeline is a text mining component that can extract competence topics from a document and generate competence records for its authors. ScholarLens reuses our Pre-processing, LODtagger and Rhetector pipelines to analyze each document. Subsequently, it attempts to find the authors of the document and create a competency record between each Author annotation in the Front Matter section and detected competence topic (named entities) found by LODtagger. In contrast to the upstream pipelines, ScholarLens generates GATE *Relation* annotations. Relations are special type of annotations within GATE that accepts two or more arguments (which refer to existing annotations with the same document). Additionally, it can provide a semantic type for the

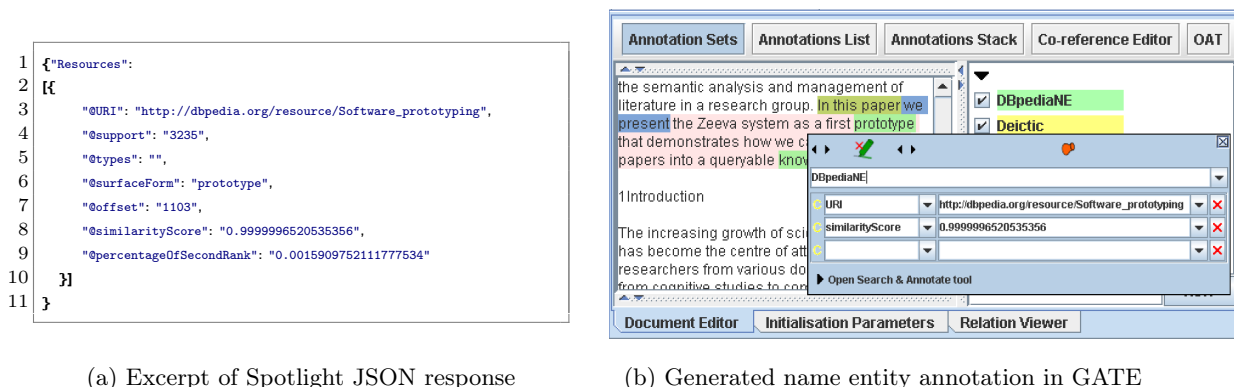


Figure 23: A JSON example response from Spotlight (left) and how the detected entity's offset is used to generate a GATE annotation in the document (right)

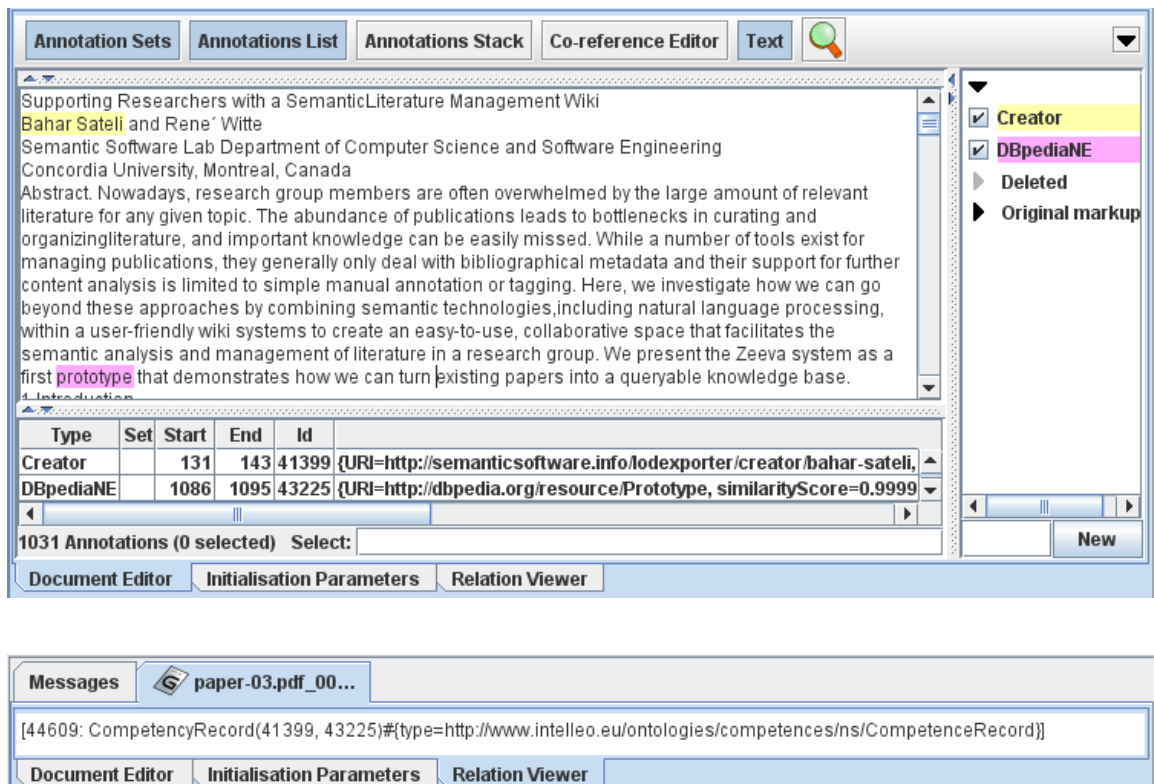


Figure 24: Annotations for an author (Creator) and a competence topic (top), and the generated competency record by ScholarLens (bottom)

relation that holds between its arguments. Figure 24 illustrates the GATE relation viewer showing the competency records generated for an example document.

7.5 Automatic Knowledge Base Population

In this section, we introduce our novel triplification component described in Section 6.4 to export NLP annotations into semantic triples.

7.5.1 LODEXporter: Flexible Generation of LOD Triples

The core idea of LODEXporter¹¹ design is that the concrete mapping, from NLP result to knowledge base, is not part of the NLP pipeline itself. Rather, it is externalized in form of a set of *mapping rules* that are encoded in the knowledge base itself. In other words, only by dynamically connecting an NLP pipeline to a knowledge base is the concrete mapping effected. The same pipeline can be

¹¹LODEXporter, <http://www.semanticsoftware.info/lolexporter>



Figure 25: The sequence of processing resources in the LODEXporter pipeline

used to export different results, using different mapping rules, to multiple knowledge bases. This is a key feature to enable agile data science workflows, where experiments with different formats can now be easily and transparently set up.

LODeXporter is a standalone GATE plugin, implemented in Java 8 based on the GATE Embedded library. It can transform GATE annotations to RDF triples based on a given mapping configuration and populate an Apache TDB-based¹² knowledge base using Jena¹³ libraries.

As mentioned earlier, the mapping rules themselves are also expressed using RDF and explicitly define which annotation types have to be exported and what vocabularies and relations must be used to create a new triple in the knowledge base. Therefore, the LODEXporter plugin is designed to read the mapping rules from the a pre-loaded knowledge base, so no additional input is required for the plugin to execute. Figure 25 shows the LODEXporter processing resource in GATE.

7.5.2 Knowledge Base Population with Document Entities

With the scholarly literature analyzed for the structural and semantic entities, we can now define a mapping configuration (see Chapter 5) for LODEXporter to export them into an interoperable and queryable knowledge base. Appendix F shows the complete mapping configuration to export the annotation results from our pipeline.

7.5.3 Knowledge Base Population with Semantic User Profiles

Similar to knowledge base population with documents, we bootstrap the user profiles with entities extracted by our ScholarLens pipeline from a set of documents (co-)authored by the user. The resulting competence records are modelled according to the mapping file shown in Appendix F and stored in the knowledge base. Note that in this step, all named entities (i.e., topics in documents and competence topics in profiles) will be merged through common LOD URIs. This approach not only prevents an excessive growth of the number of triples, but also documents and user profiles with common topics will be implicitly inter-linked.

¹²Apache TDB, <https://jena.apache.org/documentation/tdb/>

¹³Apache Jena, <https://jena.apache.org/>

7.6 An Architecture for Personal Research Agents

So far, we have described the implementation details of the components that extract pertinent information from documents and users' publications to semantically model them in a triplestore. We can now describe an end-to-end architecture for our personal research agent and demonstrate how it can exploit this workflow to offer personalized scholarly services.

7.6.1 Vector-based Representation of Scholarly Artifacts

The agent's knowledge base grows as more information are extracted from documents and user profiles in its working context. As the number of semantic triples increases, efficiently querying the large-scale knowledge becomes a challenge. While existing works try to tackle this problem through graph partitioning [Yan09] or regressing to graph databases [LPF⁺12], we propose a different approach: We construct an inverted index of the agent's knowledge base with a flexible methodology that allows us to specify what parts of the knowledge graph must be indexed. Subsequently, the agent will be able to efficiently use the index in order to create a vector-based representation of the knowledge base entities and utilize them, e.g., in a vector space model (see Section 3.2) for similarity detection between documents and users. We construct our index based on the Apache Lucene¹⁴ search engine library. We developed a GATE plugin, called *GATE2Lucene*, which can directly read GATE annotations within a document, convert them to Lucene-compatible documents and store them in an Apache Solr¹⁵ *core*. Solr is an open source, scalable and fast server implemented on top of Lucene and offers built-in ranking models, like VSM, as we explained earlier in Section 3.2. Each Solr core has a schema that organizes the index data into various fields with pre-defined types. Related fields are aggregated within a Solr *document*. We developed a Solr schema, based on the semantic representation of documents and users. Figure 26 shows an excerpt of the schema: The schema defines four fields, namely an *id* which is used to uniquely identify a document in the core, a *fulltext* that will store the entire content of the document, a *topic* field that will store the surface form of named entities found in a document, and an *entity* field that will store the LOD URIs of the said named entities. The complete schema available in Appendix H shows a similar configuration for the Contribution and Claim zones of the documents and their corresponding fields in the core. For each field, the schema specifies whether the content of each field must be analyzed, indexed or stored as-is, as well as whether each field can contain multiple values (dimensions) at once.

Additionally, we configured a set of built-in pre-processing features in Solr: All textual fields, such as *fulltext* or *contribution* sentences are tokenized, stopwords are removed and transformed

¹⁴Apache Lucene, <https://lucene.apache.org>

¹⁵Apache Solr, <http://lucene.apache.org/solr/>

```

1 < fields >
2   <field name="id" type="string" multiValued="false"
3     stored="true" indexed="true" required="true" />
4   <field name="fulltext" type="text_general" multiValued="false"
5     stored="true" indexed="true" termVectors="true" />
6   <field name="entity" type="lod" multiValued="true"
7     stored="true" indexed="true" termVectors="true" />
8   <field name="topic" type="text" multiValued="true"
9     stored="true" indexed="true" termVectors="true" />
10 </fields>
11 <uniqueKey>id</uniqueKey>

```

Figure 26: An excerpt of the Solr schema to construct semantic vectors

into lower case characters. We also apply the Snowball Porter stemmer¹⁶ on all textual fields to reduce the term vector space in the index. For fields that contain URIs, like the *entity* field, we use a regular expression filter to remove the namespace (e.g., “<http://dbpedia.org/resource/>”) from the URIs.

The GATE2Lucene plugin has a similar working mechanism to LODEXporter. It can read a configuration file at runtime that specifies what GATE annotation types and features must be written into the Solr index. The field names in the plugin configuration file must match the schema fields. The *id* field in the index must also match the document URI in the knowledge base, so they can be cross-referenced. We integrated this plugin into the agent’s workflow and present our complete architecture in Figure 27. Note that the NLP components marked with a star indicate other pipelines or components described in the previous sections.

7.6.2 Semantic Scholarly Services

As a concrete implementation of our agent’s tasks, we now demonstrate how the populated knowledge base can be used by the research agent to provide personalized services to its end-users. In particular, we show the implementation details of the functional requirements described in Section 2.1.1.

Considering the scholarly documents and user profiles in the knowledge base as artifacts of the system, we implemented each one of the above services as a task group. Each service is implemented as a sequence of tasks that may query or update the knowledge base triples. The actions corresponding to each task are formulated as parametrized SPARQL queries. There are two types of queries in our design: (i) queries looking for *concepts*, like finding all things of type `<bibo:Document>` in

¹⁶Solr Snowball Porter English stemmer, <https://wiki.apache.org/solr/LanguageAnalysis>

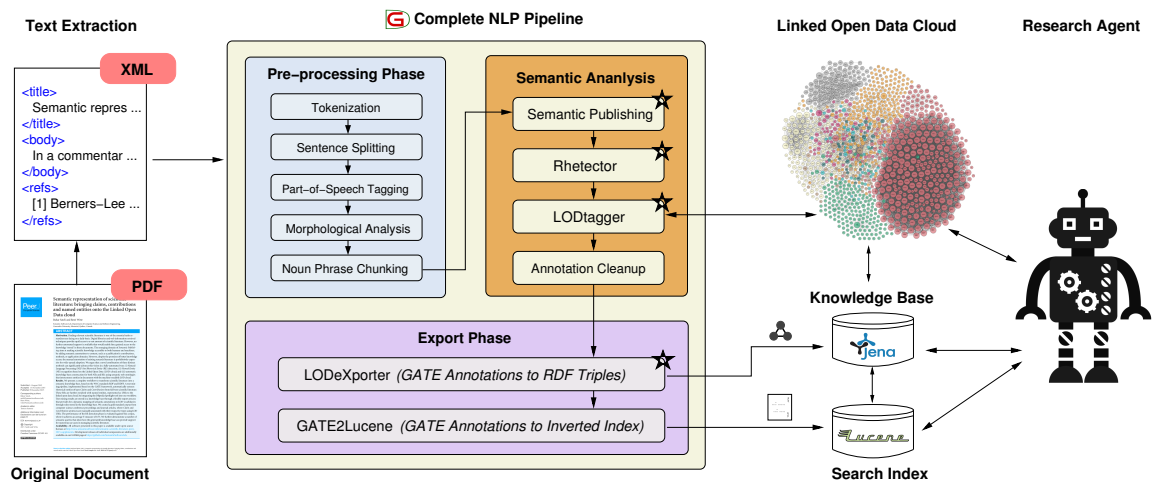


Figure 27: The complete architecture showing the end-to-end workflow for KB construction

the knowledge base, and (ii) queries that can be parameterized, such as finding all Contribution sentences mentioning ‘*linked open data*’. Wherever the required knowledge does not readily exist in the agent’s knowledge base, but may be available on the web of LOD, we incorporated federated queries to integrate additional information from external resources. In particular, we query the DBpedia ontology through its SPARQL endpoint.

Requirement #1: View a Summary of an Article

The goal of this service is to create an extractive summary from a given article by providing the end-user with the key sentences detected within the full-text of the article. Ordinarily in such a task, an end-user would have to read all of the documents she has found from a search engine in order to evaluate their relevance – a cumbersome and time-consuming task. However, our agent can automatically process the documents with its text mining pipelines, transform them into semantic triples and then query for various rhetorical types to generate an automated summary. Pseudocode 1 shows a high-level implementation of this service, referencing the query listed in Figure 28.

The agent will then show the service output in a suitable format, like the one shown in Figure 29, which dramatically reduces the amount of information that the user is exposed to, compared to a manual triage approach. Additional post-processing refinements can also be performed on the output, like visual clustering of documents by the same author, affiliations or publication date, where the metadata is available. Moreover, certain in-place substitutions, like replacing “*We*” with “*They*”, can also generate more fluent summaries.

Pseudocode 1 Automatic summary generation service

Require: A list of scholarly documents $D = \{d_1, d_2, \dots, d_n\}$ to process in a valid format (PDF, XML, etc.)

Ensure: A set of **Claim** and **Contribution** sentences from each document, an empty set otherwise

```
1: procedure GENERATESUMMARY( $D$ )
2:   for all  $d_i$  in  $D$  do                                     ▷ Knowledge extraction phase
3:     Extract the full-text of  $d_i$  using Grobid
4:     Create a GATE document from the full-text content
5:     Process the full-text with the Semantic Publishing Pipeline
6:     Process the full-text with the Rhetector Pipeline
7:     Process the full-text with the LODtagger Pipeline
8:     Export all annotations as RDF triples and store them in TDB using LODeXporter
9:     Create a Solr document from the detected entities and save to the index
10:    if  $d_i$  is co-authored by the end-user then
11:      Process the full-text with the ScholarLens pipeline
12:    for all  $d_i$  in  $D$  do                                     ▷ Output assembly phase
13:      Execute <ex:RE_query_action> using  $d_i$  URI as the parameter
14:      Execute <ex:metadata_query> using  $d_i$  URI as the parameter
15:      Return the bibliographical metadata and rhetorical entities from the queries above for each document
```

Requirement #2: Get Support in Literature Review

Retrieving document sentences by their rhetorical type still returns **Claims** or **Contributions** containing entities that are irrelevant or less interesting for an end-user in the context of a literature review task. Ideally, the agent should return only those REs that mention user-specified topics. Since we model both the REs and NEs that appear within their boundaries, the agent can allow the user to further refine her request, e.g., finding all documents that have a **Contribution** mentioning <dbpedia:Linked_data>. Pseudocode 2 shows a high-level implementation of this service, assuming

```
1 SELECT DISTINCT ?document ?content ?type WHERE {
2 {
3   ?document pubo:hasAnnotation _:contrib .
4   _:contrib rdf:type sro:Contribution .
5   _:contrib rdf:type ?type .
6   _:contrib cnt:chars ?content .
7   _:contrib oa:start ?start .
8   FILTER (?type != sro:RhetoricalElement)
9 } UNION {
10  ?document pubo:hasAnnotation _:claim .
11  _:claim rdf:type sro:Claim .
12  _:claim rdf:type ?type .
13  _:claim cnt:chars ?content .
14  _:claim oa:start ?start
15  FILTER (?type != sro:RhetoricalElement)
16 }
17 } ORDER BY ?document ?start
```

Figure 28: SPARQL query to find all Claims and Contributions within a document

SUMMARY OF DOCUMENTS IN THE SePUBICA PROCEEDINGS
Document ID: http://example.com/paper/ACL/P00-1045
<p>Title: Memory-Efficient and Thread-Safe Quasi-Destructive Graph Unification</p> <p>Authors: Marcel P. van Lohuizen</p> <p>Contributions:</p> <p>"We present a technique to reduce the memory usage of unification algorithms considerably, without increasing execution times."</p> <p>"We tested both memory usage and execution time for various configurations."</p> <p>"We reduce memory consumption of graph unification as presented in (Tomabechei, 1991) (or (Wroblewski, 1987)) by separating scratch fields from node structures."</p> <p>"We showed how to incorporate datastructure sharing."</p> <p>"Finally, we introduced deferred copying."</p> <p>Claims:</p> <p>"By assigning a different processor to each operation we obtain what we will call concurrent unification."</p> <p>"Our algorithm runs in $O(n)$ time."</p> <p>"Our algorithm allows sharing of grammar nodes, which is usually impossible in other implementations (Malouf et al., 2000)."</p> <p>"Since memory consumption is a major concern with many of the current unification-based grammar parsers, our approach provides a fast and memory-efficient alternative to Tomabechei's algorithm."</p>

Figure 29: Example entry from the agent's output in the summary generation task

the documents are already processed by the agent and relevant entities are stored in the knowledge base. By virtue of exploiting the ontologies on the LOD cloud, the agent can semantically expand the user's query to further retrieve documents that contain topics related to the named entity of interest. The federated query used in this service is listed in Figure 30.

Pseudocode 2 Literature review assistance service

Require: The knowledge base, a topic t represented by a URI

Ensure: A set of Claim and Contribution sentences from each document, an empty set otherwise

```

1: procedure FINDDIRECTMATCH( $t$ )
2:    $D \leftarrow$  Retrieve all documents with a Contribution in the knowledge base (or a subset of it)  $\triangleright$  Document retrieval phase
3:   for all  $d_i$  in  $D$  do
4:     Filter all sentences only if they contain a topic in their text resolved to the input URI
5:     Return the documents, their metadata and matched sentences
6: procedure FINDINFERREDMATCH( $t$ )
7:    $T' \leftarrow$  Retrieve all topics semantically related to  $t$  from the DBpedia ontology  $\triangleright$  Semantic query expansion phase
8:   for all  $t'_i$  in  $T'$  do
9:     FINDDIRECTMATCH( $t'_i$ )

```

The agent's output, partially shown in Table 31, is especially interesting. The agent is able to retrieve two sets of matched documents. The first set contains documents where the user's designated topic had a direct match in their full-text content. Here we can see the advantage of named entity linking on the documents: The query not only retrieved parts of articles that the user

```

1 SELECT ?document ?content ?subject {
2 {
3   SELECT DISTINCT ?document ?content WHERE {
4     ?document pubo:hasAnnotation _:contrib1 .
5     _:contrib1 rdf:type sro:Contribution .
6     _:contrib1 pubo:containsNE _:topic1 .
7     _:topic1 rdfs:isDefinedBy dbpedia:Linked_data .
8     _:contrib1 cnt:chars ?content
9   }
10 }
11 UNION
12 {
13   SELECT ?document ?content ?subject WHERE {
14     SERVICE <http://dbpedia.org/sparql> {
15       dbpedia:Linked_data <http://purl.org/dc/terms/subject> ?category .
16       ?subject <http://purl.org/dc/terms/subject> ?category .
17     }
18     ?document pubo:hasAnnotation _:contrib2 .
19     _:contrib2 rdf:type sro:Contribution .
20     _:contrib2 pubo:containsNE _:topic2 .
21     _:topic2 rdfs:isDefinedBy ?subject .
22     _:contrib2 cnt:chars ?content
23   }
24 }
25 } ORDER BY ?document

```

Figure 30: SPARQL query to retrieve all documents with a contribution related to `<dbpedia:Linked_data>`

would be interested in reading, but it also inferred that “*Linked Open Data*”, “*Linked Data*” and “*LOD*” named entities are referring to the same concept, since the DBpedia knowledge base declares an `<owl:sameAs>` relationship between their URIs in its ontology. A full-text search on the papers, on the other hand, would not have found such a semantic relation between the entities.

The second set of results provides the user with a list of documents that do not directly mention the user’s query topics, but have entities that are deemed semantically related. The agent retrieves such documents in three steps: (i) First, through a federated query to the DBpedia knowledge base, the agent finds the *category* that `dbpedia:Linked_data` is assigned to – in this case, the DBpedia knowledge base returns “*Semantic web*”, “*Data management*”, and “*World wide web*” as the categories; (ii) Then, it retrieves all other subjects which are under the same identified categories; (iii) Finally, for each related entity, the agent looks for rhetorical entities in the knowledge base that mention the related named entities within their boundaries. This way, the user receives more results from the knowledge base that cover a wider range of topics semantically related to ‘linked data’, without having to explicitly define their semantic relatedness to the system. This simple example is

DIRECTLY MATCHED DOCUMENTS dbpedia:Linked_data :
Document ID: http://example.com/paper/SePublica2014/paper-01
Title: What's in the proceedings? Combining publisher's and researcher's perspectives Authors: Volha Bryl, Aliaksandr Birukou, Kai Eckert, Mirjam Kessler Contribution: "In this paper we present a vision for having such data available as <u>Linked Open Data (LOD)</u> , and we argue that this is only possible and for the mutual benefit in cooperation between researchers and publishers."
Document ID: http://example.com/paper/SePublica2012/paper-07
Title: Linked Data for the Natural Sciences: Two Use Cases in Chemistry and Biology Authors: Cord Wiljes and Philipp Cimiano Contribution: "We present two real-life use cases in the fields of chemistry and biology and outline a general methodology for transforming research data into <u>Linked Data</u> ."
INFERRED MATCHED DOCUMENTS dbpedia:Linked_data :
Document ID: http://example.com/paper/SePublica2014/paper-05
Title: Describing bibliographic references in RDF Authors: Angelo Di Iorio, Andrea Giovanni Nuzzolese, Silvio Peroni, David Shotton, and Fabio Vitali Inferred matching topics: dbpedia:Ontology_(information_science) , dbpedia:Resource_Description_Framework Contribution: "In this paper we present two <u>ontologies</u> , i.e., BiRO and C4O, that allow users to describe bibliographic references in an accurate way, and we introduce RENhancer, a proof-of-concept implementation of a converter that takes as input a raw-text list of references and produces an <u>RDF</u> dataset according to the BiRO and C4O <u>ontologies</u> ."

Figure 31: The agent's output for assisting a researcher in a literature review task. The underlined terms in the Contribution sentences are entities semantically related to 'linked data', queried from the DBpedia ontology.

a demonstration of how the agent can exploit the wealth of knowledge available in the LOD cloud. Of course, numerous other queries now become possible on scientific papers, by exploiting other linked open data sources.

Requirement #3: Find Related Work

Researchers spend an increasingly large amount of time [CYY14] finding literature so they can compare their work against the existing body of knowledge. By leveraging its knowledge base and the information retrieval techniques we described earlier in Section 3.2, our agent can help users by finding documents with similar Claims or Contributions. Given a document or user profile, the agent can look at a set of common topics between a document/profile and the rhetorical zones of

documents within its knowledge base and construct term-vectors. Using the Vector Space Model, the agent can then compute a cosine similarity score between each pair of items and offer the top-n results to the user. The results of this service improve on keyword-based search by only retrieving documents that truly have a contribution concerning the term-vector elements, rather than merely mentioning them in their text. Figure 32 shows the agent’s output in recommending related work to user R15’s profile. As we can see in the figure, the result shown to the user is a set of contribution sentences with the matching topics visually distinguished. The results not only dramatically reduce the information our user is exposed to, but demonstrate the merit of named entity recognition by resolving words like ‘*Espresso*’ and ‘*WSD*’ to their correct word sense. This approach can be further enhanced through cross-referencing the topics in the term-vectors with the agent’s knowledge base entities, performing federated queries to the LOD cloud and expanding the vectors with semantically similar terms, like we showed in the previous service.

Requirement #4: Learn a Topic from Literature

A primary end-user group of our personal research agents are post-secondary students, using scientific literature for a variety of tasks, including learning about topics relevant to their research. While an end-user is reading a document from her reading list, the agent can provide contextual assistance whenever the user encounters a topic she has not seen before. To this end, the agent looks at the set difference between the topics in the user’s profile in the knowledge base and the named entities within the document under study. For each identified ‘new’ topic, the agent can automatically retrieve a brief description from available ontologies, or perhaps direct the user to the corresponding Wikipedia article. Additionally, it can show the user a list of documents in its knowledge base that mention the new topic to help the user understand how the topic is used in applied research works. Pseudocode 3 shows a high-level implementation of this service and an exemplary profile for researcher R_u . The SPARQL query used in this service is listed in Figure 33. The agent’s output is illustrated in Figure 34.

Pseudocode 3 Learning assistance service

Require: The knowledge base, a document d represented by a URI, a user profile u represented by a URI

Ensure: A set of new topics, their description and references, as well as relevant documents, an empty set otherwise

```

1: procedure FINDNEWTOPICS( $d$ )
2:    $T_d \leftarrow$  Retrieve all topics (named entities) in document  $d$ 
3:    $T_u \leftarrow$  Retrieve all known topics to user  $u$  from her profile
4:    $T'_u \leftarrow$  Compute the set difference between  $T_d$  and  $T_u$ 
5:   for all  $t'_i$  in  $T'_u$  do
6:     Retrieve and return the description of  $t'_i$  from the DBpedia ontology, as well as the reference to its Wikipedia page
7:   FINDDIRECTMATCH( $t'_i$ )

```

<p>Related work similar to: http://example.com/author/R15.xml</p> <p>Document Title: Graph-based Analysis of Semantic Drift in Espresso-like Bootstrapping Algorithms</p> <p>Designated topics: disambiguation, number, word, sense, vector, phase, related, convergence, graph, bootstrapping, eigenvector, matrix, algorithm, HITS, diffusion, seed, semantic, relation, kernel, iteration, von Neumann, Laplacian, instance, drift, Espresso</p>
<p>1. Document ID: http://example.com/paper/ACL/P06-1015.xml</p> <p>Title: Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations</p> <p>Contribution: "We proposed a weakly-supervised, general-purpose, and accurate <u>algorithm</u>, called <u>Espresso</u>, for harvesting binary <u>semantic relations</u> from raw text."</p>
<p>2. Document ID: http://example.com/paper/ACL/P04-1081.xml</p> <p>Title: A Kernel PCA Method for Superior Word Sense Disambiguation</p> <p>Contribution: "We introduce a new method for <u>disambiguating word senses</u> that exploits a nonlinear <u>Kernel</u> Principal Component Analysis (KPCA) technique to achieve accuracy superior to the best published individual models."</p>
<p>3. Document ID: http://example.com/paper/ACL/P06-1100.xml</p> <p>Title: Ontologizing Semantic Relations</p> <p>Contribution: "We manually built five <u>seed relation instances</u> for both <u>relations</u> and apply <u>Espresso</u> to a dataset consisting of a sample of articles from the Aquaint (TREC-9) newswire text collection."</p>
<p>4. Document ID: http://example.com/paper/ACL/P02-1044.xml</p> <p>Title: Word Translation Disambiguation Using Bilingual Bootstrapping</p> <p>Contribution: "In this paper, we propose a new method for <u>word</u> translation <u>disambiguation</u> using a <u>bootstrapping</u> technique we have developed."</p>
<p>5. Document ID: http://example.com/paper/ACL/P04-1039.xml</p> <p>Title: Relieving The Data Acquisition Bottleneck In Word Sense Disambiguation</p> <p>Contribution: "In this paper, we present an unsupervised <u>bootstrapping</u> approach for <u>WSD</u> which exploits huge amounts of automatically generated noisy data for training within a supervised learning framework."</p>

Figure 32: The agent's output in recommending related work to a user

Requirement #5: View Contributions of an Author/Group/Conference

As the agent's knowledge base scales up to accommodate an increasingly large amount of documents and their semantic entities, various time-consuming tasks, like providing an overview of a conference or a research group's contributions, can be facilitated through automated, parameterized queries. The output of this service is of interest to researchers, publishers and journal editors who would like

```

1 SELECT DISTINCT ?uri ?description ?comment WHERE {
2   ?document pubo:hasAnnotation _:re .
3   _:re rdf:type sro:RhetoricalElement .
4   _:re pubo:containsNE _:topic .
5   _:topic rdfs:isDefinedBy ?uri .
6
7   FILTER NOT EXISTS {
8     ?user rdfs:isDefinedBy <http://semanticsoftware.info/lodexporter/creator/R1> .
9     ?user um:hasCompetencyRecord _:rec .
10    _:rec c:competenceFor _:comp .
11    _:comp rdfs:isDefinedBy ?uri
12  }
13
14  SERVICE <http://dbpedia.org/sparql> {
15    ?uri rdfs:comment ?comment .
16    OPTIONAL {
17      ?uri foaf:isPrimaryTopicOf ?wiki
18    }
19  } FILTER(langMatches(lang(?comment),"en"))
20 } LIMIT 100

```

Figure 33: SPARQL query to provide learning content for topics new to researcher R_1

to obtain an overview of a conference’s proceedings or journal issue at a glance. The implementation of this service is a set of SPARQL queries for various bibliographical metadata of the documents associated with a conference, affiliation or author. Figure 35 shows an example output, providing an automatically generated overview of the ACL proceedings between 2000 and 2006.

Requirement #6: Discover Relevant New Knowledge

In order to keep researchers abreast of the latest discoveries in their field of research, the agent can proactively analyze the literature that it finds in online digital libraries and alert the researcher only when the extracted knowledge is deemed *new* to the user. For example, if a user is interested in “mobile applications” and “NLP”, the agent will only alert the user if both of her interests are within the rhetorical zones of a document. In other words, it alerts the user if there exists a paper that has a contribution on both entities of interest, and the user hasn’t read it yet. After reading the paper, the agent’s representation of the user’s knowledge becomes updated. Further occurrences of the same topic combination in a different paper would not result in a new alert, since the user already knows about it; thereby, alleviating information overload. Figure 36 shows the SPARQL query to find co-occurrence of named entities within the knowledge base documents, with respect to an end-user’s profile. The resulting output is illustrated in Figure 37.

Document ID: http://example.com/paper/PeerJ/cs-78
Title: A technology prototype system for rating therapist empathy from audio recordings in addiction
Author: Tobias Kuhn et al.
Topics New to You:
<i>Top-down and bottom-up design:</i> Top-down and bottom-up are both strategies of information processing and knowledge ordering, used in a variety of fields including [...]. (https://en.wikipedia.org/wiki/Top-down_and_bottom-up_design)
Mentions in other literature:
<ol style="list-style-type: none"> 1. "In this article, we propose to design scientific data publishing as a web-based <u>bottom-up process</u>, without top-down [...]." (http://example.com/papers/peerj/cs-78) 2. "We present a <u>bottom-up approach</u> to arranging sentences extracted for multi-document summarization." (http://example.com/paper/ACL/P06-1049) 3. "In Emam and Fisher, 2004 an example based hierarchical <u>top-down approach</u> is proposed." (http://example.com/papers/acl/P06-1073) 4. "Our work is different from Roark (2001) in that we use a <u>bottom-up</u> parsing algorithm with dynamic programming based on the parsing model II of Collins (1999)." (http://example.com/paper/ACL/P06-1030)

Figure 34: The agent's output assisting a researcher in understanding unknown topics

7.7 Summary

In this chapter, we provided the implementation details of our personal research agent's processing components, in particular the text mining pipelines developed based on the GATE framework. We showed how the knowledge extracted from literature and the user profiles are stored in a triplestore, as well as a search engine to provide the necessary information for the agent's services. Finally, we

Collection ID: http://example.com/corpus/ACL (2000–2006)
Number of documents: 600
Number of distinct authors: 935
Top 5 Prolific Authors:
<ol style="list-style-type: none"> 1. Daniel Marcu (Information Sciences Institute, Department of Computer Science, University of Southern California, USA) 2. Mark Johnson (Brown University Providence, USA) 3. Jianfeng Gao (Natural Language Computing Group Microsoft Research Asia, China) 4. Ming Zhou (Natural Language Computing Group Microsoft Research Asia, China) 5. Jian Su (Institute for Infocomm Research, Singapore)
Number of distinct affiliations: 1,264 from 39 countries
Top 5 Prolific Countries: USA, Japan, Germany, UK, China

Figure 35: The agent's output providing an overview of a corpus


```

1 SELECT DISTINCT ?content1 ?uri1 ?uri2 WHERE {
2   ?document pubo:hasAnnotation _:re1 .
3   _:re1 rdf:type sro:RhetoricalElement .
4   _:re1 cnt:chars ?content1 .
5   _:re1 pubo:containsNE ?topic1 .
6   ?topic1 rdfs:isDefinedBy ?uri1 .
7   _:re1 pubo:containsNE ?topic2 .
8   ?topic2 rdfs:isDefinedBy ?uri2 .
9   FILTER(?topic1 != ?topic2)
10
11  FILTER EXISTS {
12    ?user rdfs:isDefinedBy <http://semanticsoftware.info/lolexporter/creator/R1> .
13    ?user um:hasCompetencyRecord _:rec1 .
14    _:rec1 c:competenceFor _:comp1 .
15    _:comp1 rdfs:isDefinedBy ?uri1 .
16
17    ?user um:hasCompetencyRecord _:rec2 .
18    _:rec2 c:competenceFor _:comp2 .
19    _:comp2 rdfs:isDefinedBy ?uri2 .
20    FILTER(?uri1 != ?uri2)
21  }
22 } LIMIT 10

```

Figure 36: SPARQL query to find documents with a novel combination of topics interesting for researcher R_1

showed how the fusion of information within the knowledge base, combined with the LOD cloud, can provide novel, semantic scholarly services to assist a variety of end-user groups in their tasks.

New Knowledge Alert: Co-occurrence of dbpedia:Mobile_device & dbpedia:Web_service
Document ID: http://example.com/paper/MobiWIS2013/paper-09
<p>Title: Design and Development Guidelines for Real-Time, Geospatial Mobile Applications: Lessons from 'MarineTraffic'</p> <p>Authors: Dimitrios Zissis, Dimitrios Lekkas, and Panayiotis Koutsabasis</p> <p>Contribution: "In this paper we present the case of the design and development of the <u>mobile</u> version of MarineTraffic (marinetraffic.com), which is an example of a real-time, geospatial, community-based <u>web service</u> that allows users to view vessel information, positions, routes and port traffic in real-time."</p>

Figure 37: The agent's output in issuing an alert on discovering new knowledge

Chapter 8

Evaluation

In this chapter, we delve into the details of how we evaluated the separate components of our agent’s design, in particular the text mining pipelines and the generated knowledge base.

8.1 Semantic Publishing Pipeline Evaluation

The effectiveness of our agent’s functional requirements, like summarization of articles (Requirement #1) or finding related work (Requirement #3), is directly impacted by how well the text mining pipelines are able to extract relevant entities from scholarly documents. Towards this end, we assess our semantic publishing pipeline described in Section 7.2, in terms of its precision and recall in finding structural and bibliographical entities, i.e., we apply as an *intrinsic* evaluation approach.

8.1.1 Gold standard

At the core of an intrinsic evaluation is a gold standard dataset that has the correct results (annotations, features, classes, and attributes), manually annotated by human domain experts. We make use of the gold standard corpora from two years of an international semantic publishing competition. The *Semantic Publishing Challenge* (SemPub) is a series of international competitions, co-located with the Extended Semantic Web Conference (ESWC), with the aim of producing and exploiting contextual information extracted from scientific literature in the computer science domain. In particular, Task 2 of the challenge series focuses on extracting various metadata from the PDF full-text of papers and answering a set of queries using NLP and Named Entity Recognition techniques. The competition is conducted in two separate phases: In the *training* phase, a dataset (PDF files) are provided to the participants, as well as the corresponding gold standard output. Once the training phase is over, the *evaluation* phases commences, during which the final evaluation dataset is

provided, but the gold standard is not released until the competition is over. Conforming to the competition’s rules, the extracted information must be stored in a knowledge base and participants must submit the populated knowledge base, as well as the query results. The official evaluator tool of the competition then compares the submissions against the gold standard, which is in form of Comma Separated Values (CSV) files, and calculates the relevant metrics for the competition. We participated in two editions of the above competition: In the SemPub Challenge 2015,¹ we submitted an automatic workflow for knowledge base population with a subset of the CEUR-WS² proceedings, for which we received the “*Most Innovative Approach*” award; In the SemPub Challenge 2016,³ we submitted an improved workflow and obtained the second highest F_1 -measure of 0.63 among 5 groups and 17 researchers who participated in the competition, only 0.14 less than the best performing tool.

For our evaluations in this section, we had to perform some data transformations, as the gold standard CSV files were not directly useable within GATE. Our goal was to use GATE’s *Corpus Quality Assurance* plugin [CMB⁺11] to calculate Precision, Recall and F_1 -measure of our pipelines in various settings. Therefore, we first retrieved the source PDF files of the training and evaluation datasets, scraped their plain full-text and manually annotated them with the correct annotations from the CSV files within the GATE Developer environment. Table 13a shows the size of the Semantic Publishing Challenge corpora, with the number of documents for each dataset and the average number of sentences per document. Table 13b provides the number of each annotation type in the corresponding datasets of SemPub2016. Unfortunately, the gold standard corpus for SemPub2015 is only partially available at the time of this writing.⁴

8.1.2 Results

We executed our semantic publishing pipeline described in Section 7.2 on the gold standard corpora and calculated the Precision, Recall, and F_1 -score of each dataset.

Tables 14a and 14b, respectively, show the evaluation results on the training and evaluation datasets of the SemPub 2016 challenge. The reported metrics are the average number calculated by the GATE’s Corpus Quality Assurance tool, allocating a half weight to partially correct responses (see Section 3.1).

As we can see from the table, the pipeline performs best in detecting float elements, followed by detection of authors in both datasets. The detection of affiliations in both datasets suffers due to a large number of overlapping annotations with the gold standard, which discounts the weight of each partially correct annotation by half. We further analyzed the overlapping annotations in both

¹Semantic Publishing Challenge 2015, <https://github.com/ceurws/lod/wiki/SemPub2015>

²CEUR Workshop Proceedings, <http://ceur-ws.org/>

³Semantic Publishing Challenge 2016, <https://github.com/ceurws/lod/wiki/SemPub2016>

⁴The SemPub2015 gold standard corpus only covers 50 query results. See https://github.com/ceurws/lod/wiki/SemPub15_Task2.

Table 13: Statistics of the semantic publishing challenge gold standard corpora

(a) Size of the challenge datasets

Year	Training Dataset		Evaluation Dataset	
	#Documents	Avg. #Sentences	#Documents	Avg. #Sentences
2016	45	371	40	307
2015	101	616	82	703

(b) Number of annotations in the SemPub 2016 gold standard corpus

Dataset	#Annotations						Total
	Author	Affiliation	Location	Section	Figure	Table	
Training	149	80	61	304	173	56	823
Evaluation	117	51	46	275	110	37	636
Total	266	131	107	579	283	93	

datasets and found out that faulty line breaks introduced by the text extraction tool that we used (PDFX [CPV13]) was accountable for several annotations, causing a loss in the F_1 -measure of our pipeline.

As per rules of the competition, we added the LODeXporter plugin to our semantic publishing pipeline, in order to populate a TDB-based knowledge base with the extracted annotations. For SemPub 2016, we targeted only 5 out of 8 queries of the competition, as the remaining queries were concerned with finding European project names and funding agencies. Figure 38 shows an example query from the competition and our results. In this example, the author’s affiliations do not have any explicit country name in their adjacent text. However, our Affiliation-Location inferrer (see Section 6.1.3) ran the university’s name (e.g., “McGill University”) against the DBpedia ontology and retrieved the subject of a triple matching `<dbpedia:McGill_University,dbpedia:country,?subject>`.

We used the competition’s official evaluator tool to calculate the precision and recall of the knowledge base in answering the queries. The total number of queries was 360 (45 papers per query) and 320 (40 papers per query) for the training and evaluation phases, respectively. Tables 15a and 15b show the evaluation results of the knowledge base in answering queries, based on the triples generated from the training and evaluation datasets of SemPub2016.

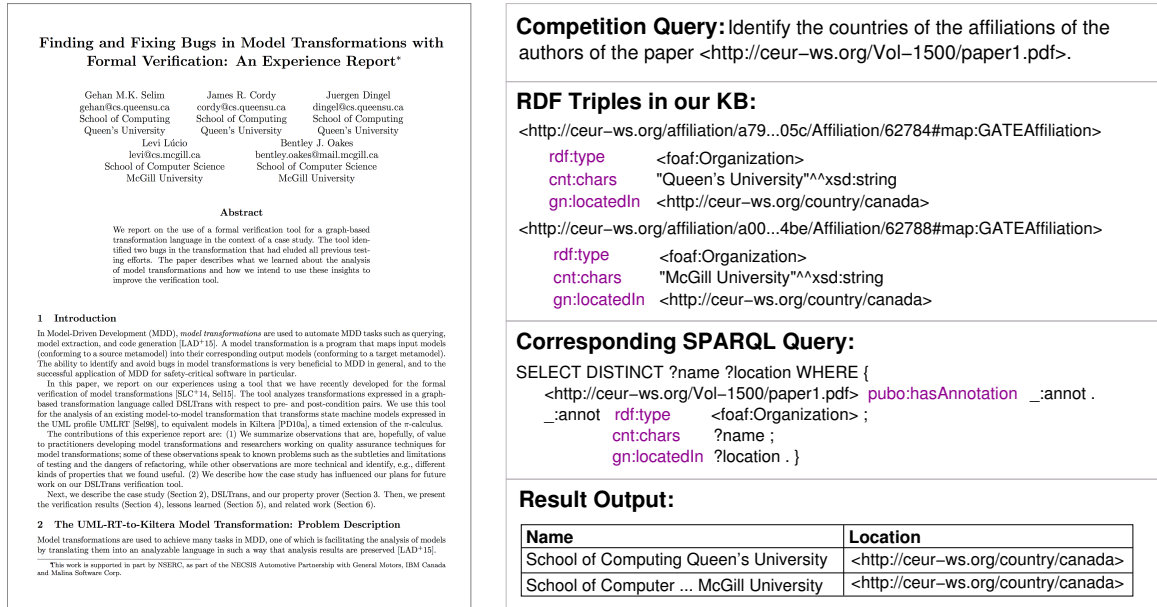


Figure 38: Example query from the semantic publishing challenge and our query results

8.2 Rhetector Pipeline Evaluation

Value-added services, such as finding related work (Requirement #3) or providing an overview (Requirement #5), are made possible when the agent is capable of understanding the scientific contributions of each document and the argumentations of its authors. To this end, we developed Rhetector (see Section 7.3.1) that can extract rhetorical entities of a document on a sentential level and LODtagger (see Section 7.3.2) to annotate all documents with domain topics. In this section, we evaluate the accuracy of these pipelines against a gold standard corpus.

8.2.1 Gold standard

In Section 4.1.1, we reviewed some of the related works on rhetorical analysis of scholarly literature. Unlike the semantic publishing challenge explained in the previous section, there exists no widely-accepted gold standard corpora for the evaluation of our discourse analysis pipelines. Rather, several disparate corpora are available from different domains, such as the chemistry or biomedical domain. Therefore, we had to first assemble a gold standard corpus of literature from the computer science domain. Our emphasis here was to annotate the full-text of open access documents from the computer science, software engineering and related areas. The resulting corpus contains the following sets:

PeerJCompSci is a collection of 22 open-access papers from the computer science edition of the

Table 14: Evaluation results of our pipeline on the Semantic Publishing Challenge gold standard

(a) Intrinsic evaluation on the SemPub 2016 training gold standard corpus

Annotation	#Match	#Only GS	#Only Resp.	#Overlap	Precision	Recall	F ₁ -measure
Affiliation	20	31	28	35	0.45	0.44	0.44
Author	94	30	5	8	0.92	0.74	0.82
Figure	133	11	1	14	0.95	0.89	0.92
Location	53	10	26	0	0.67	0.84	0.75
Section	180	80	47	4	0.79	0.69	0.74
Table	37	5	5	2	0.86	0.86	0.86
Macro Summary	n/a	n/a	n/a	n/a	0.77	0.74	0.75
Micro Summary	517	167	112	63	0.79	0.73	0.76

(b) Intrinsic evaluation on the SemPub 2016 evaluation gold standard corpus

Annotation	#Match	#Only GS	#Only Resp.	#Overlap	Precision	Recall	F ₁ -measure
Affiliation	12	18	21	35	0.43	0.45	0.44
Author	92	18	5	5	0.93	0.82	0.87
Figure	100	5	3	4	0.95	0.94	0.94
Location	29	12	0	0	1.00	0.71	0.83
Section	201	66	18	3	0.91	0.75	0.82
Table	35	2	0	1	0.99	0.93	0.96
Macro Summary	n/a	n/a	n/a	n/a	0.87	0.77	0.81
Micro Summary	469	121	47	48	0.87	0.77	0.82

PeerJ journal,⁵ which were annotated by us for this dissertation.

SePublica corpus contains 29 documents from the proceedings of the Semantic Publishing workshops⁶ from 2011–2014, which were annotated by us for this dissertation.

AZ is a collection of 80 conference articles in computational linguistics, originally curated by Teufel [Teu99].⁷

Each sentence containing a rhetorical entity was manually annotated (by the author of this dissertation) and classified as either a Claim or Contribution by adding the respective class URI from the SRO ontology as the annotation feature. The documents in these corpora are in PDF or XML

⁵PeerJ Computer Science Journal, <https://peerj.com/computer-science/>

⁶Semantic Publishing Workshop (SePublica), <http://sepublica.mywikipaper.org/drupal/>

⁷Argumentation Zoning (AZ) Corpus, http://www.cl.cam.ac.uk/~sht25/AZ_corpus.html

Table 15: Evaluation results of the semantic publishing challenge 2016 queries

(a) Intrinsic evaluation of queries over the training dataset

Query	Description	Precision	Recall	F ₁ -measure
Q1	Identify the affiliations of the authors of the paper X.	0.71	0.56	0.61
Q2	Identify the countries of the affiliations of the authors in the paper X.	0.69	0.64	0.66
Q4	Identify the titles of the first-level sections of the paper X.	0.62	0.59	0.60
Q5	Identify the captions of the tables in the paper X.	0.85	0.85	0.85
Q6	Identify the captions of the figures in the paper X.	0.71	0.68	0.69
Overall		0.71	0.66	0.68

(b) Intrinsic evaluation of queries over the evaluation dataset

Query	Description	Precision	Recall	F ₁ -measure
Q1	Identify the affiliations of the authors of the paper X.	0.78	0.74	0.74
Q2	Identify the countries of the affiliations of the authors in the paper X.	0.55	0.54	0.54
Q4	Identify the titles of the first-level sections of the paper X.	0.74	0.72	0.73
Q5	Identify the captions of the tables in the paper X.	0.76	0.76	0.76
Q6	Identify the captions of the figures in the paper X.	0.81	0.80	0.80
Overall		0.72	0.71	0.71

formats, and range from 3–43 pages in various styles (ACM, LNCS, and PeerJ). Table 17 shows the statistics of our gold standard corpus.

8.2.2 Results

We divided our evaluation methodology into two separate parts: First, we assess the Rhetector pipeline in terms of its precision, recall and F₁-measure against the gold standard. Later in this section, we also conduct an evaluation experiment using LODtagger.

We executed our Rhetector pipeline on our manually-annotated gold standard corpus and used GATE’s Corpus QA tool to calculate the relevant metrics. Table 18 shows the results of our evaluation. On average, the Rhetector pipeline obtained a 0.73 F₁-measure on the evaluation dataset. We gained some additional insights into the performance of Rhetector by manually investigating the pipeline’s output. When comparing the AZ and SePublica corpora, we can see that the pipeline achieved almost the same F-measure for roughly the same amount of text, although the two datasets are from different disciplines: SePublica documents are semantic web-related workshop papers,

Table 16: Evaluation results of the semantic publishing challenge 2015 queries

Query	Description	Precision	Recall	F ₁ -measure
Q1	Identify the affiliations of the authors of the paper X.	0.80	0.56	0.66
Q2	Identify the papers presented at the workshop X and written by researchers affiliated to an organization located in the country Y.	0.80	0.53	0.64
Q3	Identify all works cited by the paper X.	0.75	0.54	0.63
Q4	Identify all works cited by the paper X and published after the year Y.	0.40	0.15	0.22
Q5	Identify all journal papers cited by the paper X.	1.00	0.00	0.00
Q6	Identify the grant(s) that supported the research presented in the paper X (or part of it).	1.00	0.00	0.00
Q7	Identify the funding agencies that funded the research presented in the paper X (or part of it).	0.20	0.20	0.20
Q8	Identify the EU project(s) that supported the research presented in the paper X (or part of it).	1.00	0.00	0.00
Q9	Identify the ontologies mentioned in the abstract of the paper X.	0.29	0.50	0.36
Q10	Identify the ontologies introduced in the paper X (according to the abstract).	0.11	0.30	0.16
Overall		0.335	0.277	0.274

whereas the AZ corpus contains conference articles in computational linguistics. Another interesting observation is the robustness of Rhetector’s performance when the size of an input document (i.e., its number of tokens) increases. For example, when comparing the AZ and PeerJ CompSci performance, we observed only a 0.05 difference in the pipeline’s (micro) F-measure, even though the total number of tokens to process was doubled (42,254 vs. 94,271 tokens, respectively).

An error analysis of the intrinsic evaluation results showed that the recall of our pipeline suffers when: *(i)* the authors’ contribution is described in passive voice and the pipeline could not attribute it to the authors, *(ii)* the authors used unconventional metadiscourse elements; *(iii)* the rhetorical entity was contained in an embedded sentence; and *(iv)* the sentence splitter could not find the

Table 17: Statistics of the discourse analysis gold standard corpora

Dataset	Training Dataset		#Annotations	
	#Documents	Avg. #Sentences	Claim	Contribution
PeerJ CompSci	22	530.6	110	126
SePublica	29	340.3	62	163
AZ	10	212.1	19	44

Table 18: Results of the intrinsic evaluation of Rheteor

Dataset	Detected REs		Precision	Recall	F ₁ -measure
	Claims	Contributions			
PeerJ CompSci	73	226	0.67	0.80	0.73
SePublica	55	168	0.81	0.80	0.80
AZ	22	44	0.76	0.79	0.78
Average	50	146	0.75	0.80	0.77

correct sentence boundary, hence the rhetorical entity annotations span covered more than one sentence.

Additionally, we extended the dataset used in the evaluation above and processed all the documents with the LODtagger pipeline to investigate the populated knowledge base. Table 19 shows the quantitative results of the populated knowledge base. The total number 1.08 million RDF triples were in the knowledge base. On average, the processing time of extracting REs, NEs, as well as the triplification of their relations was 5.55, 2.98 and 2.80 seconds per document for the PeerJCompSci, SePublica and AZ corpus, respectively; with the DBpedia Spotlight annotation process taking up around 60% of the processing time (running on a standard 2013 quad-core desktop PC).

For each corpus, we ran a number of queries on the knowledge base to count the occurrences of NEs and REs in the contained documents. The ‘DBpedia Named Entities (Occurrences)’ column shows the total number of NEs tagged by Spotlight, whereas the ‘DBpedia Named Entities (Distinct URIs)’ column shows the total of named entities with a unique URI. For example, if we have both ‘linked open data’ and ‘LOD’ tagged in a document, the total occurrence would be two, but since they are both grounded to the same URI (i.e., <dbpedia:Linked_data>), the total distinct number of NEs is one. This is particularly interesting in relation to their distribution within the documents’ rhetorical zones (column ‘Distinct DBpedia NE/RE’). As can be seen in Table 19, the number of NEs within REs are an order of a magnitude smaller than the total number of distinct named entities throughout the whole papers. This holds across the three distinct corpora we evaluated.

This experiment shows that NEs are not evenly distributed in scientific literature. Overall, this is encouraging for our hypothesis that the combination of NEs with REs brings added value, compared to either technique alone: As mentioned in the example above, a paper could mention a topic, such as ‘Linked Data’, but only as part of its motivation, literature review, or future work. In this case, while the topic appears in the document, the paper does not actually contain a contribution involving linked data. Relying on standard information retrieval techniques hence results in a large amount of noise when searching for literature with a particular contribution. Semantic queries on

Table 19: Quantitative analysis of the populated knowledge base: We processed three datasets for REs and NEs. The columns ‘Distinct URIs’ and ‘Distinct DBpediaNE/RE’ count each URI only once throughout the KB, hence the total is not the sum of the individual corpora, as some URIs appear across them.

Dataset	Size		DBpedia Named Entities		Rhetorical Entities		Distinct DBpediaNE/RE	
	Docs	Sents	Occurrences	Distinct URIs	Claims	Contributions	Claims	Contributions
PeerJCompSci	27	15928	58808	8504	92	251	378	700
SePublica	29	8459	31241	4915	54	165	189	437
AZ	80	16803	74896	6992	170	463	563	900
Total	136	41,190	164,945	14,583	316	879	957	1,643

the other hand, as we propose them here, can easily identify relevant papers in a knowledge base.

8.3 Semantic User Profiling User Study

We performed two rounds of evaluations to assess the accuracy of our generated user profiles: We first performed a pilot study with a small number of researchers. We then studied the incorrect entries in the generated user profiles and refined our approach for an extended user study.

In our first evaluation round, we reached out to ten computer scientists from Concordia University, Canada and the University of Jena, Germany (including the author of this dissertation) and asked them to provide us with a number of their selected publications. We processed the documents and populated a knowledge base with the researchers’ profiles. Using a Java command-line tool that queries this knowledge base, we generated \LaTeX documents as a human-readable format of the researchers’ profiles, each listing the top-50 competence topics, sorted by their occurrence in the users’ publications. For each participant, we exported two versions of their profile: (i) a version with a list of competences extracted from their papers’ full-text, and (ii) a second version that only lists the competences extracted from the rhetorical zones of the documents, in order to test our hypothesis described in Section 6.3. Subsequently, we asked the researchers to review their profiles across two dimensions: (i) the relevance of the extracted competences and (ii) their level of expertise for each extracted competence. To ensure that none of the competence topics are ambiguous to the participants, our command-line tool also retrieves the English label and comment of each topic from the DBpedia ontology using its public SPARQL⁸ endpoint. The participants were instructed to choose only one level of expertise among (‘Novice’, ‘Intermediate’, ‘Advanced’) for each competence and select ‘Irrelevant’ if the competence topic was incorrect or grounded to a wrong sense. Figure 39

⁸DBpedia SPARQL endpoint, <http://dbpedia.org/sparql>

Table 20: Evaluation results for the generated user profiles in the first user study: This table shows the number of distinct competence topics extracted from the ten participants and the average precisions at 10, 25 and 50 cut-off ranks. The last row (MAP) shows the mean average precision of the system at various cut-offs.

ID	#Docs	#Distinct Competences		Avg. Precision@10		Avg. Precision@25		Avg. Precision@50	
		Full Text	REs Only	Full Text	REs Only	Full Text	REs Only	Full Text	REs Only
R1	8	2,718	293	0.91	0.80	0.84	0.74	0.80	0.69
R2	7	2,096	386	0.95	0.91	0.90	0.92	0.87	0.91
R3	6	1,200	76	0.96	0.99	0.93	0.95	0.92	0.88
R4	5	1,240	149	0.92	0.92	0.86	0.81	0.77	0.75
R5	4	1,510	152	0.84	0.99	0.87	0.90	0.82	0.82
R6	6	1,638	166	0.93	1.0	0.90	0.97	0.88	0.89
R7	3	1,006	66	0.70	0.96	0.74	0.89	0.79	0.86
R8	8	2,751	457	0.96	1.0	0.92	1.0	0.92	0.99
R9	9	2,391	227	0.67	0.73	0.62	0.70	0.56	0.65
R10	5	1,908	176	0.96	0.91	0.79	0.80	0.69	0.70
MAP				0.88	0.92	0.83	0.87	0.80	0.81

shows an entry from a generated profile in our pilot study. A complete profile example is included in the supplementary materials of this dissertation.

SEMANTIC USER PROFILING EVALUATION SHEET

NAME: _____

For each topic, please choose **only one** of the available options that best represents your level of expertise:

Novice means “I am somewhat familiar with this topic.”
Intermediate means “I have conducted research on this topic and feel competent in it.”
Advanced means “I am an expert in this topic.”

#	Competency Topic	Novice	Intermediate	Advanced	Irrelevant
1	Recommender system <i>Recommender systems or recommendation systems (sometimes replacing "system" with a synonym such as platform or engine) are a subclass of information filtering system that seek to predict the 'rating' or 'preference' that user would give to an item. Recommender systems have become extremely common in recent years, and are applied in a variety of applications. The most popular ones are probably movies, music, news, books, research articles, search queries, social tags, and products in general.</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 39: An automatically generated user profile in L^AT_EX format

The evaluation results of our pilot study are shown in Table 20. In this study, a competence was considered as relevant when it had been assigned to one of the three levels of expertise. For each participant, we measured the average precision (see Section 3.1) of the generated profiles in both the full-text and RE-only versions. The results show that for both the top-10 and top-25 competences,

70–80% of the profiles generated from RE-only zones had a higher precision, increasing the system Mean Average Precision (MAP) up to 4% in each cut-off. In the top-50 column, we observed a slight decline in some of the profiles’ average precision, which we believe to be a consequence of more irrelevant topics appearing in the profiles, although the MAP score stays almost the same for both versions.

8.3.1 User Study: Error Analysis

In order to refine our approach, we went back to the users from the first study to understand the root cause of competences they marked as *irrelevant*. We classified each irrelevant competence into one of four error categories:

Type 1 (Wrong URI). The profile contains a wrong URI: This is typically caused by the linking tool (see LODtagger in Section 7.3.2) assigning the wrong URI to a surface form; either because it picked the wrong sense among a number of alternatives or the correct sense does not exist in the knowledge base.

Type 2 (Empty description). As explained above, we retrieve the *comment* for each competence URI to make sure users understand their profile entries. In about 3% of profile entries, this automatic process failed, leading to an empty description, which was often marked as irrelevant. We identified three main causes for this: (a) a timeout in the SPARQL query to the public DBpedia endpoint; (b) a missing comment entry in English for some resources in the online DBpedia; and the much rarer cause (c) where the URI generated by the linking tool was valid for an older version of DBpedia, but has meanwhile been removed.

Type 3 (User misunderstanding). Some users interpreted the task differently when it came to identifying their competences: Rather than evaluating what they are generally competent in, they marked each entry that did not fall into their research fields as irrelevant. For example, a researcher working on web services marked ‘HTTP (Protocol)’ as irrelevant, since HTTP was not a research topic in itself, though the user clearly had knowledge about it.

Type 4: (Unspecific competence). Users often assigned ‘irrelevant’ for competences that were deemed too broad or unspecific. The cause is very similar to Type 3, with the main difference that competences here were high-level concepts, like ‘*System*’, ‘*Idea*’, or ‘*Methodology*’, whereas Type 3 errors were assigned to technical terms, like ‘*HTTP*’, ‘*Data Set*’, or ‘*User (computing)*’.

The results of this analysis are summarized in Table 21. As can be seen, the majority of the errors (77% and 82%) are of Type 1. This is consistent with earlier observations we had about DBpedia

Table 21: Error analysis of the irrelevant competence entries generated for the participants in the first user study: For each error type, the total numbers of irrelevant competences in the profile and its percentage (rounded) is shown.

		User							
	Error Type	R8	R1	R6	R9	R3	R4	R2	Average
Full-text profiles	Type 1	4	7	10	7	6	16	7	8.14
		100%	64%	91%	30%	100%	89%	100%	82%
	Type 2	0	2	0	1	0	0	0	0.43
		0%	18%	0%	4%	0%	0%	0%	3%
	Type 3	0	0	0	8	0	0	0	1.14
		0%	0%	0%	35%	0%	0%	0%	5%
	Type 4	0	2	1	7	0	2	0	1.71
		0%	18%	9%	30%	0%	11%	0%	10%
RE profiles	Type 1	1	13	10	13	14	14	5	10
		50%	65%	100%	45%	93%	88%	100%	77%
	Type 2	0	1	0	2	0	0	0	0.43
		0%	5%	0%	7%	0%	0%	0%	2%
	Type 3	0	3	0	11	1	1	0	2.29
		0%	15%	0%	38%	7%	6%	0%	9%
	Type 4	1	3	0	3	0	1	0	1.14
		50%	15%	0%	10%	0%	6%	0%	12%

Spotlight when applying it to research literature. Modifying or retraining Spotlight itself was out of the scope of this work, but we addressed some common errors in our pipeline, as described below.

8.3.2 Extended Experiments

With the lessons learned from our first experiment, we enhanced our ScholarLens pipeline to remove the error types iterated in the previous section. In particular, to address Type 1 errors, we excluded exporting entities with surface forms like “*figure*” or “*table*” from newly generated profiles, as these were consistently linked to irrelevant topics like “*figure painting*” or “*figus*”. To address Type 3 and Type 4 errors, we refined the task description shown to participants before they start their evaluation. Additionally, we introduced a competence classification to distinguish general competences from technical and research competences. However, to accommodate this classification we dropped the previous assessment of competence levels, as we did not want to double the workload of our study participants.

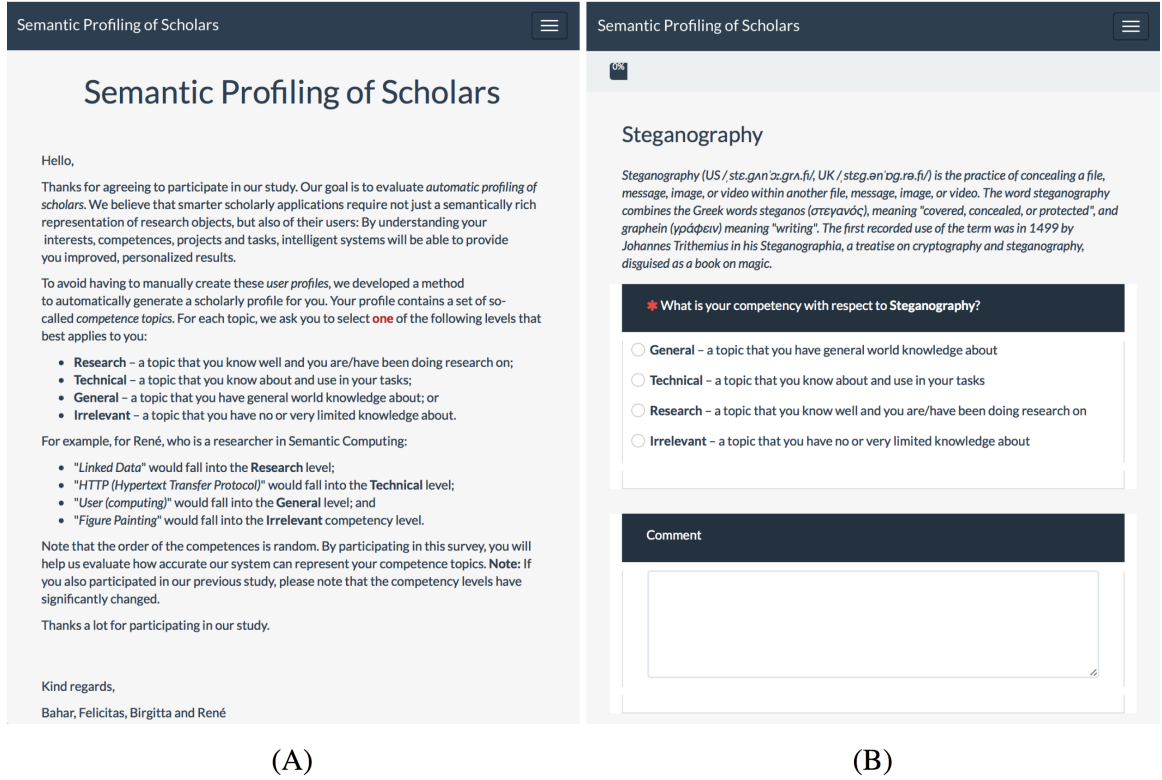


Figure 40: An automatically generated web-based survey using *LimeSurvey*: (A) depicts the instructions shown to participants to explain the survey motivation and how to select a competence level, and (B) shows an example competency question with an interactive response interface.

Automatic Generation of Online Surveys

For our revised experiment, we set up a web-based user profile evaluation system. In the new set up, instead of generating \LaTeX profiles for users, we implemented a survey-style profile generation tool that queries the populated knowledge base and generates web-based profiles compatible with LimeSurvey,⁹ an open source survey application with built-in analytics features, as shown in Figure 40. Similar to the first experiment, we generated two surveys for each user: One with the competence topics extracted from the full-text of documents and one with topics extracted from the rhetorical zones only. To lessen the *priming bias* [Lav08] – where participants may think topics shown earlier in the survey must be more relevant to them – we randomized the order of survey questions and informed the users in the evaluation instructions about this fact. However, we internally kept the original rank of the competence topics shown in survey questions as they appear in the knowledge base profiles, so that we can compute the precision of our system in top- k cut-off ranks. We invited 32 computer scientists to participate in our user evaluations (excluding the author

⁹LimeSurvey, <https://www.limesurvey.org>

of this dissertation). In total, 25 users responded to the survey (note that an anonymized user like ‘R1’ from the second study is not necessarily the same person as in the first study). The populated knowledge base from the user profiles contains 4.7 million triples, referring to about 15 thousand unique competence topics. In contrast to the previous survey, this time we asked the users to rate the competences along three different competence types, namely *General* which comprises very general and broad topics such as ‘*System*’ or ‘*Number*’, *Technical* which refers to skills a computer scientist needs in daily work, e.g., ‘*Hypertext Transfer Protocol*’, and *Research*, which points to research topics a user has been or currently is involved in, e.g., ‘*Linked Data*’. A complete web-based survey is included in the supplementary materials of this dissertation.

Result Computation

All responses were exported into comma-separated value (CSV) files and analyzed with our own Java-based command-line tool (see Appendix A), transforming the original horizontal schema into a vertical structure, based on their original rank. We computed the Precision@Rank, the Mean Average Precision (MAP) and the normalized Discounted Cumulative Gain (nDCG), according to the equations presented in Chapter 3. Table 22 presents the responses for both versions, the full-text profiles and RE Zones, with respect to the overall ratings across the four different competence levels. The results for the precision metrics are displayed in Tables 23 and 24.

Since Precision@k and MAP are based on binary ratings (relevant/non-relevant), it has to be specified which competence levels to take into account. Therefore, we defined two thresholds: *Irrelevant* (Threshold ‘0’) and *General* (Threshold ‘1’). For threshold ‘0’, we treated the responses in the *General*, *Technical* and *Research* competence types as relevant. In this case, only *Irrelevant* entries are counted as errors. However, this might not be appropriate for every application: some use cases might want to also exclude competences in the *General* category. Therefore, we also analyzed the results for ratings above *General*, in order to ensure an equal distribution (Tables 23 and 24). Here, competences were only considered as relevant when they were rated either as *Technical* or *Research*.

Additionally, we computed the nDCG for each profile, which does not penalize for irrelevant competence topics in profiles.

Discussion

Overall, compared to our first user study, our enhanced method resulted in fewer irrelevant results in the user profiles. This is partially due to the improvements mentioned above, where we removed irrelevant entries that affected every generated profile (e.g., the competence entries derived from the word “*figure*”).

We also analyzed the distribution of competence topic types in each user profile (Figure 41).

Table 22: Analysis of the survey responses for profiles generated from Full-text and RE Zones. The values shown in the columns are the number of competence types as voted by the survey participants.

User	Competence Type in Full-text				Competence Type in RE Zones			
	General	Technical	Research	Irrelevant	General	Technical	Research	Irrelevant
R1	19	7	13	11	17	10	8	15
R2	6	21	18	5	9	21	15	5
R3	14	16	12	8	9	9	9	23
R6	24	8	15	3	12	10	13	15
R9	13	18	9	10	7	25	6	12
R10	12	23	13	2	18	18	5	9
R11	17	19	13	1	20	17	9	4
R12	12	16	21	1	19	13	16	2
R14	11	23	10	6	16	19	6	9
R15	12	23	8	7	16	18	3	13
R16	14	18	15	3	15	22	10	3
R17	16	16	12	6	20	18	8	4
R18	5	12	30	3	15	22	13	0
R19	8	20	15	7	9	14	18	9
R21	3	9	36	2	4	13	32	1
R23	22	18	8	2	18	18	9	5
R25	10	23	10	7	14	15	12	9
R26	18	15	8	9	22	9	6	13
R27	16	14	13	7	12	19	13	6
R28	2	27	18	3	4	24	18	4
R29	6	8	22	14	6	15	12	17
R30	13	21	12	4	22	6	7	15
R31	9	19	14	8	14	14	11	11
R35	7	7	31	5	5	19	18	8
R36	17	9	17	7	9	9	17	15
Total								
306 410 393 141 332 397 294 227								
24.48% 32.80% 31.44% 11.28% 26.56% 31.76% 23.52% 18.16%								

In both profile versions, about 55-65% of the detected competences were rated either as *Technical* or *Research*, which corroborates our hypothesis that the named entities in users' publications are representative of their research expertise. In comparison with the full-text and RE-only version of each user profile, although we observe an increase in the number of irrelevant topics, the majority of them fall into the *Research* and *Technical* types. These results are also consistent with our hypothesis that the topics in rhetorical zones of scholarly literature, like claims and contributions of authors, are strong indications of their competence. As we can see from the results, the full-text profiles returned

Table 23: Precision computation for profiles generated from full-text with a relevance threshold of *Irrelevant (0)* and *General (1)*. All ratings above *Irrelevant (0)* and *General (1)* have been considered as relevant, respectively.

User	Threshold 0 - Irrelevant						Threshold 1 - General						nDCG
	Average Precision			Precision@k			Average Precision			Precision@k			
	@10	@25	@50	@10	@25	@50	@10	@25	@50	@10	@25	@50	
R1	0.90	0.87	0.85	0.80	0.88	0.78	0.88	0.69	0.63	0.60	0.52	0.40	0.90
R2	0.87	0.86	0.88	0.80	0.88	0.90	0.75	0.79	0.80	0.70	0.84	0.78	0.88
R3	1.00	0.90	0.88	1.00	0.84	0.84	0.88	0.80	0.75	0.90	0.68	0.56	0.92
R6	1.00	0.99	0.96	1.00	0.96	0.94	0.37	0.45	0.45	0.40	0.44	0.46	0.83
R9	0.95	0.89	0.85	0.90	0.80	0.80	0.70	0.63	0.58	0.60	0.52	0.54	0.80
R10	1.00	1.00	0.99	1.00	1.00	0.96	0.96	0.89	0.84	0.9	0.8	0.72	0.93
R11	1.00	1.00	1.00	1.00	1.00	0.98	0.90	0.81	0.76	0.80	0.72	0.64	0.92
R12	1.00	1.00	1.00	1.00	1.00	0.98	0.79	0.78	0.78	0.70	0.84	0.74	0.88
R14	0.93	0.94	0.94	0.90	0.96	0.88	0.93	0.88	0.82	0.90	0.80	0.66	0.88
R15	0.82	0.80	0.82	0.70	0.80	0.86	0.83	0.70	0.64	0.60	0.60	0.62	0.87
R16	1.00	1.00	0.98	1.00	1.00	0.94	1.00	0.90	0.83	0.9	0.8	0.66	0.95
R17	1.00	0.94	0.91	0.90	0.92	0.88	0.83	0.73	0.66	0.60	0.64	0.56	0.89
R18	1.00	0.98	0.97	1.00	0.96	0.94	0.79	0.86	0.86	0.90	0.88	0.84	0.94
R19	1.00	0.96	0.91	1.00	0.88	0.86	0.82	0.72	0.70	0.70	0.64	0.70	0.88
R21	1.00	0.96	0.95	0.90	0.96	0.96	1.00	0.96	0.94	0.90	0.96	0.90	0.97
R23	0.99	0.96	0.96	0.90	0.96	0.96	0.61	0.62	0.60	0.70	0.56	0.52	0.84
R25	0.93	0.86	0.85	0.90	0.84	0.86	0.92	0.81	0.75	0.80	0.72	0.66	0.89
R26	0.93	0.92	0.90	0.90	0.88	0.82	0.67	0.58	0.55	0.50	0.52	0.46	0.84
R27	0.81	0.83	0.86	0.80	0.84	0.86	0.77	0.68	0.63	0.70	0.52	0.54	0.86
R28	1.00	0.97	0.94	1.00	0.88	0.94	1.00	0.97	0.94	1.00	1.00	0.88	0.97
R29	0.92	0.83	0.79	0.80	0.72	0.72	0.75	0.70	0.67	0.70	0.6	0.6	0.86
R30	0.91	0.89	0.91	0.90	0.92	0.92	0.54	0.60	0.64	0.50	0.68	0.66	0.85
R31	0.95	0.93	0.89	0.90	0.92	0.84	0.71	0.72	0.71	0.70	0.76	0.66	0.87
R35	0.79	0.88	0.90	0.90	0.88	0.86	0.77	0.80	0.79	0.80	0.76	0.76	0.90
R36	0.99	0.91	0.88	0.90	0.88	0.86	0.99	0.91	0.88	0.90	0.88	0.86	0.94
	Mean Average Precision			Average			Mean Average Precision			Average			
	0.95	0.92	0.91	0.91	0.91	0.89	0.80	0.76	0.73	0.74	0.70	0.66	0.89

less irrelevant results and higher ratings (64%) than the RE-only version (55%). A closer look on individual responses revealed that the error Type 1 (Wrong URI) occurred more often in the RE-only version. A wrong matching of extracted terms to URIs mainly causes a wrong description and hence an irrelevant result. Longer and more comprehensive text passages, as in the full-text profiles, might better compensate this problem and therefore result in less URI mismatches. Too broad and general competences are a further issue when looking at the ratings. Again, the reason was that DBpedia Spotlight that does not distinguish between composite and single terms, for instance, “*service*” and “*service provider*”. It finds successful matches for both terms and thus produces general topics.

We also evaluated the rated competences with respect to their ranking in the result list. Both

Table 24: Precision computation for profiles generated from RE zones with a relevance threshold of *Irrelevant* (0) and *General* (1). All ratings above *Irrelevant* (0) and *General* (1) have been considered as relevant, respectively.

User	Threshold 0 - Irrelevant						Threshold 1 - General						nDCG
	Average Precision			Precision			Average Precision			Precision			
	@10	@25	@50	@10	@25	@50	@10	@25	@50	@10	@25	@50	
R1	0.93	0.90	0.86	0.90	0.80	0.70	0.84	0.68	0.56	0.50	0.40	0.36	0.85
R2	0.87	0.83	0.86	0.70	0.88	0.90	0.86	0.83	0.79	0.70	0.80	0.72	0.88
R3	0.99	0.93	0.87	0.90	0.80	0.54	0.98	0.87	0.81	0.80	0.60	0.36	0.90
R6	0.98	0.91	0.83	0.90	0.76	0.70	0.60	0.60	0.55	0.60	0.52	0.46	0.83
R9	0.91	0.80	0.79	0.70	0.80	0.76	0.79	0.66	0.64	0.60	0.64	0.62	0.81
R10	0.99	0.94	0.90	0.90	0.88	0.82	0.83	0.75	0.62	0.70	0.48	0.46	0.87
R11	1.00	1.00	0.95	1.00	0.96	0.92	0.99	0.86	0.71	0.80	0.56	0.52	0.90
R12	1.00	1.00	0.99	1.00	1.00	0.96	0.67	0.60	0.60	0.50	0.56	0.58	0.88
R14	1.00	0.97	0.90	1.00	0.84	0.82	0.72	0.71	0.64	0.80	0.60	0.5	0.87
R15	0.99	0.90	0.84	0.90	0.80	0.74	0.70	0.65	0.59	0.70	0.56	0.42	0.82
R16	1.00	0.99	0.96	1.00	0.92	0.94	0.95	0.88	0.76	0.90	0.64	0.64	0.91
R17	1.00	1.00	0.98	1.00	0.96	0.92	0.88	0.81	0.71	0.80	0.64	0.52	0.90
R18	1.00	1.00	1.00	1.00	1.00	1.00	0.86	0.80	0.73	0.80	0.68	0.70	0.93
R19	1.00	0.99	0.93	1.00	0.88	0.82	0.75	0.70	0.69	0.50	0.68	0.64	0.89
R21	1.00	1.00	1.00	1.00	1.00	0.98	1.00	1.00	0.99	1.00	1.00	0.9	0.95
R23	1.00	0.97	0.95	1.00	0.92	0.90	0.78	0.76	0.71	0.80	0.68	0.54	0.88
R25	1.00	0.88	0.85	0.80	0.84	0.82	0.98	0.77	0.69	0.70	0.60	0.54	0.93
R26	0.98	0.88	0.80	0.90	0.72	0.74	0.61	0.53	0.41	0.40	0.28	0.30	0.80
R27	1.00	0.97	0.93	1.00	0.88	0.88	0.95	0.89	0.79	0.90	0.68	0.64	0.93
R28	0.84	0.84	0.87	0.90	0.84	0.92	0.84	0.83	0.82	0.90	0.76	0.84	0.90
R29	0.60	0.66	0.67	0.60	0.72	0.66	0.53	0.58	0.56	0.50	0.64	0.54	0.76
R30	0.82	0.81	0.77	0.70	0.76	0.70	0.83	0.53	0.45	0.30	0.36	0.26	0.84
R31	0.96	0.86	0.81	0.80	0.72	0.78	0.64	0.57	0.51	0.50	0.48	0.50	0.81
R35	1.00	0.88	0.84	1.00	0.88	0.84	1.00	0.92	0.83	1.00	0.76	0.74	0.92
R36	1.00	0.94	0.85	0.90	0.84	0.70	0.95	0.86	0.77	0.90	0.88	0.86	0.94
	Mean Average Precision			Average			Mean Average Precision			Average			
	0.95	0.91	0.88	0.90	0.86	0.82	0.82	0.74	0.68	0.70	0.62	0.57	0.88

metrics, Precision@k and Mean Average Precision have been computed across two relevance thresholds. Threshold ‘0’ denotes the results where all ratings above *Irrelevant* were considered as relevant, namely, *General*, *Technical* and *Research*. Since this division favours relevant competences, we additionally computed the Precision@k and Mean Average Precision for ratings above *General*. Among the top-10 results, the RE-only profiles performed slightly better for Threshold ‘1’, which indicates that all the relevant competences are a bit more likely in the Top-10 results than in the full-text profiles. However, the ranking results turn upside down for the Top-50 results, where the MAP value for the full-text version is significantly higher than for the RE-only version. That reveals the ranking in full-text profiles is more stable over 50 competences, compared to RE zones.

Additionally, we analyzed whether the different number of papers per user has an influence on

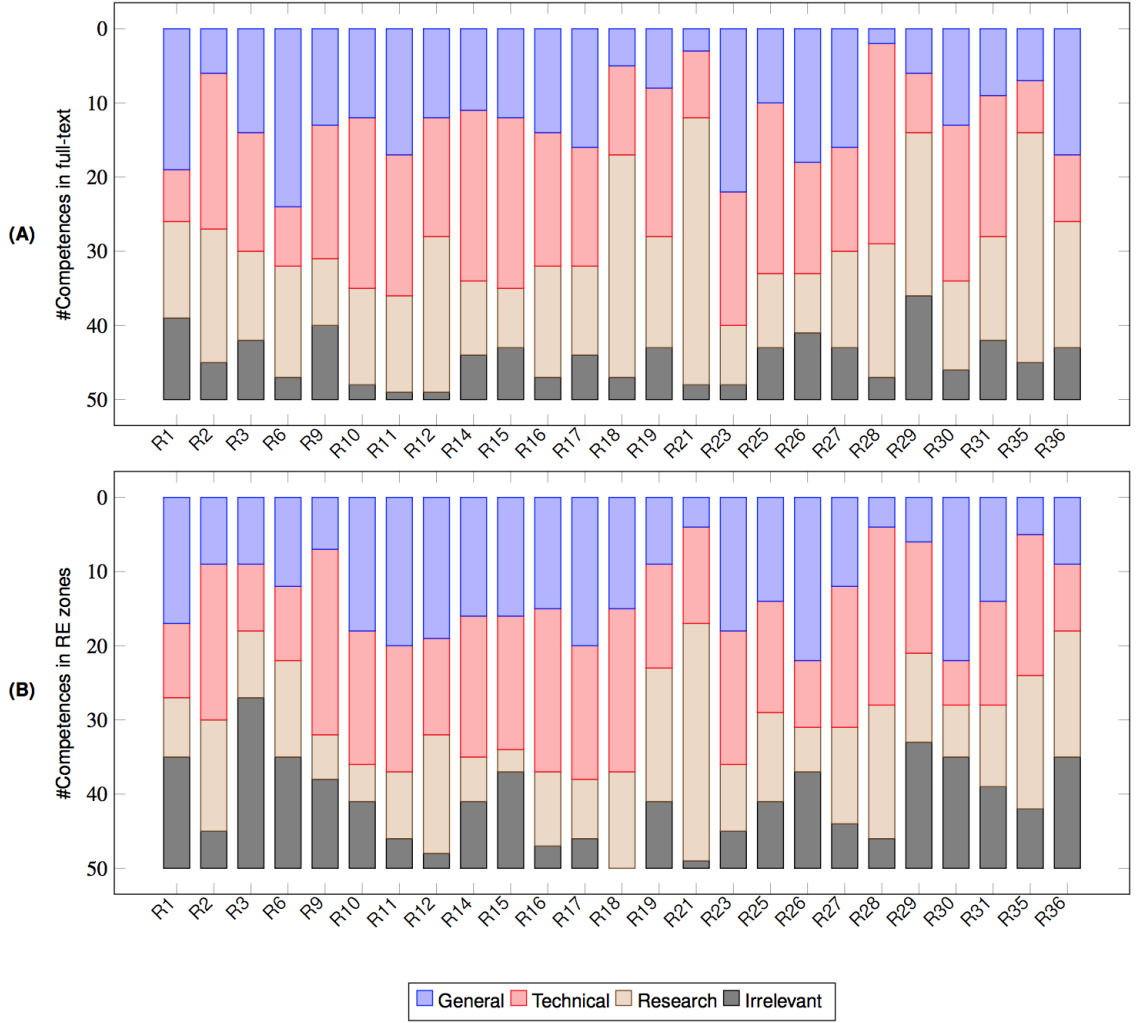


Figure 41: The two plots show the distribution of top-50 competence types in full-text (A) and RE-only (B) profiles from the evaluation survey responses.

the results. It turned out that there is no correlation between the number of papers used and the obtained precision (see Appendix A).

Overall, we reached high ranking values in the top-10 results for both thresholds, the MAP values for *Research* and *Technical* competences vary between 0.80 and 0.95. Looking at the Top-50 competences, full-text profiles performed better than RE zones. Hence, we can conclude that our approach is effective for detecting a user’s background knowledge implicitly. Table 25 shows the summary of our extended user study.

Note that, we neither take into account a decay function nor distinguish between recent publications or papers a user has written a longer time ago. This leaves room for future work. For all our results, it also needs to be considered that we did not ask the users about missing competences

Table 25: Summary of the scholarly user profiling evaluations

Scope	Threshold 0 - Irrelevant						Threshold 1 - General						Avg. nDCG
	Mean Avg. Precision			Avg. Precision			Mean Avg. Precision			Avg. Precision			
	@10	@25	@50	@10	@25	@50	@10	@25	@50	@10	@25	@50	
Full-text	0.95	0.92	0.91	0.91	0.91	0.89	0.80	0.76	0.73	0.74	0.70	0.66	0.89
RE Zone	0.95	0.91	0.88	0.90	0.86	0.82	0.82	0.74	0.68	0.70	0.62	0.57	0.88

and therefore we did not penalize for incomplete results. The results are grounded on relevance assessments for automatically extracted competences from publications. Rather than completeness, it is an approach to counteract the cold-start problem in personalized applications and to minimize the burden for the user to explicitly provide pertinent information.

8.4 Semantic Vectors Evaluation

Throughout the previous sections, we evaluated how well we can extract structural and semantical elements of scholarly documents to populate the knowledge base and bootstrap user profiles. Here, we further examine whether a vector representation of these semantic entities can be used by the agent to measure user-item and item-item similarity for scalable recommendation and personalization purposes.

8.4.1 Gold standard

The focus of our evaluation in this section is whether the agent can find a set of relevant documents for a user, in services like finding relevant work (Requirement #3). We reuse an existing gold standard corpus from Sugiyama et al.’s work in [SK10]. The gold standard represents 15 junior researchers and their respective set of relevant results from the proceedings of the Annual Meetings of the Association for Computational Linguistics (ACL) between 2000 and 2006. Each junior researcher has one published paper that we use to bootstrap his profile and subsequently utilize to find matching documents. The list of relevant documents for each researcher is also included in the gold standard, available as a set of document IDs from the ACL Anthology¹⁰ collection available online, like the example shown in Table 26. The gold standard corpus contains 600 articles, each of which is 8 pages long and has an average length of 300 sentences.

8.4.2 Experiment Design

The assessment of our agent’s recommendation capabilities is a complex issue, as multiple factors can impact the construction of semantic vectors from the knowledge base. To make our evaluation

¹⁰ACL Anthology, <https://aclanthology.info>

Table 26: Example entry from the recommendation gold standard dataset

Researcher	Researcher's Publication DOI	Relevant Document IDs in the ACL Corpus
R1	doi: 10.1145/1378889.1378921	P00-1002, P00-1044, P00-1062, P00-1064, P01-1026, P01-1064, P03-1028, P03-1029, P04-1056, P05-1058

approach tractable, we break down our work to answer the following questions:

Q1 Which document entities have the most distinguishing effect on the vector-based similarity computations?

Q2 What is the best configuration for the construction of term-vectors in Solr?

Q3 What is the highest average precision that our agent can achieve in a recommendation task?

Q4 Which term-vector construction technique provides the best ranking in the result set?

To answer Q1, we chose 7 different combination of document entities to conduct our experiment:

1. All words within the full-text of a document (considered as the baseline)
2. Surface forms of the named entities (topics) in a document
3. Named entity URIs in a document, dereferenced from an ontology
4. Verbatim content of the **Contribution** sentences in a document
5. Surface forms of the named entities within **Contribution** sentences in a document
6. Named entity URIs within **Contribution** sentences, dereferenced from an ontology
7. Combination of 4, 5 and 6

For each of the above combinations, we execute our recommendation techniques and compare its performance by computing an average precision (see Section 3.2.2) over all researchers from the gold standard.

As described earlier, we use the VSM methodology to find similar items in the index (and hence, the knowledge base). Apache Solr allows us to configure the minimum *tf* and *idf* values when constructing term-vectors. In other words, we can determine the significance of each index term by defining thresholds for how frequent and rare they have to be in the entire term/document space, respectively. To answer Q2, we chose two intervals of [0..10] for both *tf* and *idf* values and repeated our recommendation techniques for each of the 100 possibilities to find the best configuration. By using the best performing configuration, we report the highest average precision obtained (Q3) when we create semantic vectors of the entities determined by Q1. Finally, we look at how well each of our recommendation techniques can rank the results set in the top 5 and 10 documents retrieved in a recommendation service, by computing the average normalized discounted cumulative gains (see Section 3.2.2) over all researchers, answering Q4.

8.4.3 Results

We processed the gold standard corpus using the architecture shown in Figure 27 and populated (i) a TDB-based triplestore and (ii) a Lucene-based index with the results. It takes an average of 4 seconds per paper to analyze its full-text and generate RDF triples according to the mapping file. The populated knowledge base contains 3.32 million triples in total, comprising 0.54 million named entities describing 19,153 distinct topics, 935 distinct authors associated with 1,264 distinct affiliations from 39 countries, as well as 5,059 **Contribution** and 2,029 **Claim** sentences. We also repeated the same process with the researchers’ publications, resulting in 15 additional pseudo-documents in the index; Each new document represents the profile of the respective researcher identified by R_n .

We use Solr’s ‘*More Like This*’ (MLT) component to retrieve the top 5 and 10 documents similar to each researcher’s profile. MLT constructs term vectors from user-defined *interesting* fields and uses Okapi BM25F [MRS08] to rank the results. We queried the index 700 times (100 tf-idf combinations for 7 techniques) and calculated the precision by comparing the retrieved document IDs with the gold standard set for each researcher and reported the average. The complete set of resulting diagrams are available in Appendix G and the best performing combination is shown in Figure 42. We observed across all combinations that semantic vectors constructed from the surface forms of named entities (topics) in the full-text of the document consistently performed above the baseline, increasing the average precision by 8%, as reported in Table 27. We found out that the highest average precision of our best performing configuration uses $df = 6$ and $tf = 3$. The term-vectors constructed from the document topics obtained an average precision of 0.39 over all researchers in retrieving the 10 most similar documents. As for the rhetorical entities in a document, we determined configuration 7 (the combination of text and topics within the **Contribution** zone of a document) to be effective, but performing about 5% below the baseline in the top-5 documents. The average precision, however, drops significantly in the top-10, most likely due to the fact that Rhetector could not find any rhetorical entities in the remaining documents. These experiments answer our questions Q1–Q3.

Finally, we use the best performing configuration and look at the relevance of the retrieved documents for each researcher to calculate the nDCG. The results are shown in Table 28 for the top-5 and top-10 documents. We can observe that semantic vectors created based on named entity surface forms (topics) and URIs (entities) provide the highest nDCG, beating the baseline by up to 13%. As for the rhetorical zones, the average nDCG obtained from the RE zones also showed to be higher when looking at the **Contribution** parts of a document, where available. This final experiment answers Q4 and concludes the semantic vector-based evaluation of our personal research agent.

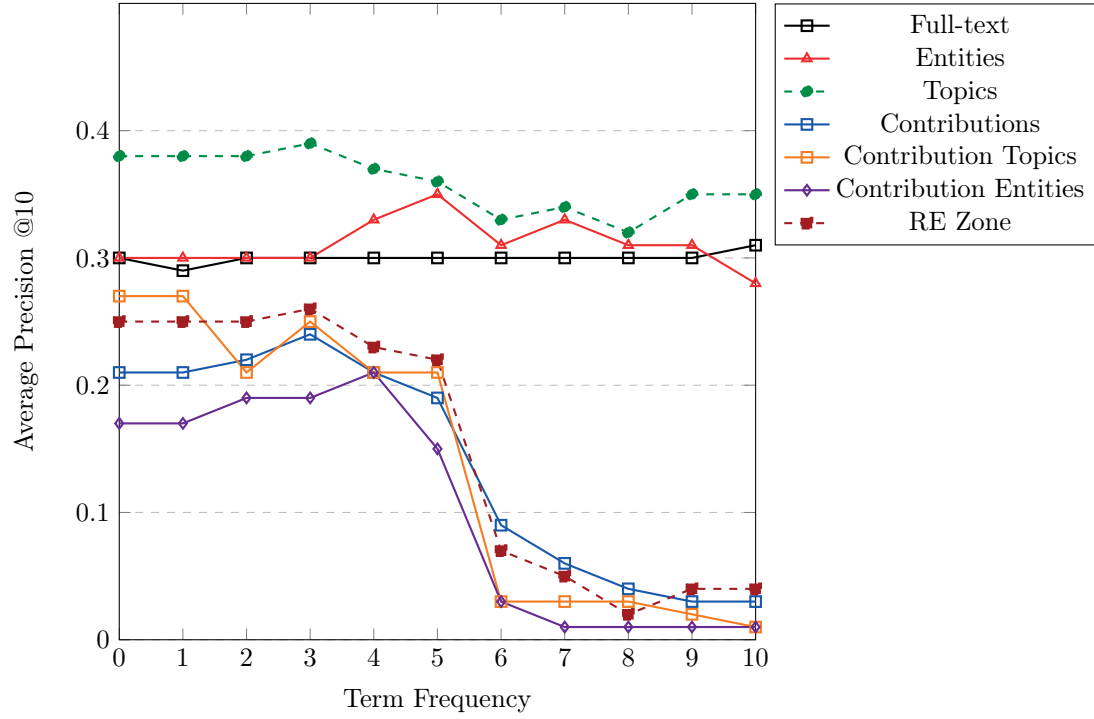


Figure 42: Best performing configuration for document recommendation ($df = 6$)

8.5 Summary

In this chapter, we delineated the evaluation methodologies we used to assess the performance of our agent’s various components. We described the precision and recall of our text mining pipelines in extracting relevant information from scholarly documents to store in the knowledge base and populate end-user profiles. We showed that the agent’s bootstrapping mechanism to construct scholarly user profile in a non-intrusive way resulted in highly accurate representations of the users’ background knowledge. Finally, we showed that our agent can construct semantic vectors of entities, both from the document models and user profiles in the knowledge base and leverage them to recommend articles to users in a given task.

Table 27: Average precision for various recommendation configurations

Researcher ID	Full-text	Entities	Topics	Contribs.	Contrib. Topics	Contrib. Entities	RE Zone
R1	0.10	0.00	0.10	0.00	0.00	0.00	0.00
R2	0.00	0.00	0.10	0.00	0.00	0.00	0.00
R3	0.00	0.30	0.20	0.30	0.60	0.70	0.40
R4	0.00	0.00	0.10	0.00	0.00	0.20	0.00
R5	0.10	0.10	0.10	0.10	0.10	0.10	0.00
R6	0.20	0.00	0.00	0.20	0.10	0.00	0.20
R7	0.50	0.60	0.50	0.60	0.60	0.50	0.60
R8	1.00	0.80	0.80	0.50	0.00	0.10	0.30
R9	0.70	0.60	0.80	0.20	0.80	0.50	0.70
R10	0.00	0.40	0.50	0.40	0.50	0.10	0.40
R11	0.40	0.30	0.30	0.30	0.00	0.00	0.20
R12	0.40	0.30	0.60	0.20	0.30	0.30	0.40
R13	0.30	0.20	0.40	0.00	0.00	0.00	0.00
R14	0.60	0.30	0.60	0.20	0.00	0.00	0.00
R15	0.30	0.60	0.70	0.60	0.80	0.40	0.70
Average	0.31	0.30	0.39	0.24	0.25	0.19	0.26

Table 28: Average nDCG for various recommendation configurations

Researcher ID	Full-text	Entities	Topics	Contribs.	Contrib. Topics	Contrib. Entities	RE Zones
R1	0.17	0.34	0.00	0.00	0.00	0.00	0.00
R2	0.00	0.00	0.00	0.00	0.00	0.00	0.00
R3	0.00	0.17	0.21	0.51	0.53	0.49	0.53
R4	0.00	0.00	0.00	0.00	0.00	0.00	0.00
R5	0.00	0.34	0.00	0.00	0.00	0.34	0.00
R6	0.00	0.00	0.00	0.00	0.21	0.00	0.13
R7	0.52	0.49	0.66	0.28	0.45	0.36	0.32
R8	1.00	0.87	1.00	0.52	0.00	0.00	0.13
R9	0.66	1.00	0.79	0.51	1.00	0.53	1.00
R10	0.00	0.64	0.47	0.35	0.51	0.00	0.55
R11	0.21	0.13	0.34	0.34	0.00	0.00	0.35
R12	0.34	0.85	0.49	0.35	0.51	0.55	0.52
R13	0.13	0.17	0.36	0.00	0.00	0.00	0.00
R14	0.66	0.70	0.34	0.34	0.00	0.00	0.00
R15	0.72	0.64	1.00	0.68	0.85	0.51	0.87
Average @5	0.29	0.42	0.38	0.26	0.27	0.19	0.29

Researcher ID	Full-text	Entities	Topics	Contribs.	Contrib. Topics	Contrib. Entities	RE Zone
R1	0.11	0.22	0.00	0.00	0.00	0.00	0.00
R2	0.00	0.07	0.00	0.00	0.00	0.00	0.00
R3	0.00	0.17	0.29	0.40	0.56	0.60	0.42
R4	0.00	0.08	0.00	0.00	0.00	0.15	0.00
R5	0.00	0.22	0.07	0.08	0.07	0.22	0.00
R6	0.14	0.00	0.00	0.07	0.14	0.00	0.09
R7	0.54	0.60	0.57	0.53	0.51	0.45	0.49
R8	1.00	0.84	0.79	0.47	0.00	0.08	0.23
R9	0.65	0.87	0.66	0.33	0.87	0.48	0.80
R10	0.00	0.57	0.45	0.37	0.54	0.07	0.51
R11	0.28	0.15	0.36	0.37	0.00	0.00	0.22
R12	0.44	0.63	0.38	0.22	0.40	0.42	0.41
R13	0.22	0.32	0.23	0.00	0.00	0.00	0.00
R14	0.63	0.66	0.35	0.29	0.00	0.00	0.00
R15	0.47	0.70	0.72	0.66	0.83	0.46	0.78
Average @10	0.30	0.41	0.32	0.25	0.26	0.20	0.26

Chapter 9

Conclusions

The unprecedented rate of scientific output is a major threat to the productivity of knowledge workers, who rely on scrutinizing the latest scientific discoveries for their daily tasks. This issue not only overwhelms any manual efforts of researchers in combing through the vast amount of knowledge available in ever-growing digital repositories, but also affects learners and academic publishers alike. In this dissertation, we aspired to create a *Personal Research Agent* that can provide personalized services to knowledge workers for reading, writing and learning from scientific literature. This is a particularly challenging objective, since the information that needs to be processed into machine-readable knowledge is written in a natural language, and large in scale. In this work, we showed how a confluence of techniques from the Natural Language Processing, Semantic Web and Information Retrieval domains can realize this novel vision. In this chapter, we revisit our research contributions and suggest some directions for future work.

9.1 Summary

The contributions of this work are manifold: We presented a complete architecture built entirely based on open standards and open source tools that we developed to support a research agent’s workflow in semantic analysis of scholarly artifacts.

Semantic Modeling of Scholarly Documents. Due to the use of natural languages, like English, as a means of written communication between scholarly authors, the semantic analysis of scientific publications becomes a complex task. Despite a number of existing de-facto standards for argumentation structures in scientific articles, authors use diverse rhetorical moves to present their work to the readers. This is further exacerbated by the ambiguity of words in a natural language. An agent needs to have access to a formalized ‘understanding’ of documents’ content with explicit,

unambiguous semantics, in order to evaluate their relevance when fulfilling a user’s task. In this dissertation, we designed multiple semantic models for the description of bibliographical, structural and semantical elements of scholarly literature that can describe authorship information, arguments and key topics, using the W3C recommended Resource Description Framework (RDF) and RDF Schema. We developed a flexible, fault-tolerant text mining methodology to extract bibliographical metadata from scientific articles for their categorical management and retrieval and showed how they can be interlinked with external resources on the Web. We also developed a text mining solution to detect two types of rhetorical entities, namely *Claims* and *Contribution* sentences, from the full-text of documents and semantically interlink their type and embedded topics to resources on the Linked Open Data (LOD) cloud. We published our semantic modelling of scientific articles at the SAVE-SD workshop at the World Wide Web conference in 2015 and won the ‘*Best Paper Award*’. Finally, we utilized our text mining pipelines for document modeling to participate at the international Semantic Publishing Challenge 2016, Task 2, and obtained the second highest F-measure in the competition.

Semantic Modeling of Scholarly Users. What distinguishes our research agent from a semantically-enhanced digital library is its ability to offer the end-users scholarly services that fulfill specific information needs, such as drafting a literature review, and adapt their output based on the background knowledge of a user. All adaptation techniques require a user profiling method that can provide sufficient information about a user’s context. In this dissertation, we designed a semantic model for a formal description of scholarly users, in particular, their background knowledge and interests, in form of a set of competency records. We showed that we can bootstrap these user profiles by examining the publication history of a researcher in a non-intrusive way and overcome the infamous ‘cold-start’ problem. Similar to the document model, we devised a controlled vocabulary for scholarly semantic user profiles that links to existing linked open data resources on the Web. Through common topics and competency records, the agent can implicitly measure item-item and user-item similarities for recommendation purposes or personalization of its services, e.g., by showing only those articles that may provide ‘new’ knowledge for a user. We also demonstrated that these profiles can accurately represent scholarly users’ competences in the context of two user studies.

A Novel Approach for Automatic Knowledge Base Construction. An essential component of our research agent’s design is a *knowledge base* – an adaptable, graph-based representation of scholarly users and documents. Since manually populating such a knowledge base with the large amount of available information is prohibitively expensive, we developed an automatic approach that uses natural language processing techniques to mark up relevant information in a document (or user profile), combined with a highly flexible, innovative process that can transform them into semantic

Table 29: Mapping of our research goals with their resulting contributions and publications

Research Goal	Contributions	Publications
Goal 1: Design a Semantic Scholarly KB	<ul style="list-style-type: none"> • Semantic model of documents' rhetorical and named entities • PUBlication Ontology (PUBO) • Semantic model of scholarly user profiles • Two user studies for scholarly profiling 	[10, 11, 21, 22]
Goal 2: Automatic Construction of the KB	<ul style="list-style-type: none"> • An open-source workflow for semantic publishing experiments • Semantic Publishing 2015 & 2016 (pipelines used for bibliographical metadata detection) • Rhetector (rule-based RE extraction) • LODtagger (NER in scholarly literature) • ScholarLens (automatic scholarly user profiling) • LODEXporter (flexible RDF generation component) 	[11, 20, 21, 23, 24, 29]
Goal 3: Design of a Personal Research Agent	<ul style="list-style-type: none"> • Semantic model of a personal research agent • Personal Research Agents vocabulary (PRAV) • Semantic vector-based recommendation of scholarly literature • Implementation of various personalized scholarly services • Development of a prototypical user interface (Zeeva) 	[6, 17, 18, 19, 25, 26]

triples in RDF format. Designed based on ‘separation of concerns’ principles, the *triplication* process can be customized by a knowledge engineer or end-user using a declarative language. At the same time, it relieves a language engineer from dealing with generating unique identifiers for entities in a document or selecting a semantic vocabulary for describing them. Essentially, our approach provides an agile solution for the construction of LOD-compliant knowledge bases for different application scenarios. Our solution, including the LODEXporter component, was awarded as the ‘*Most Innovative Approach*’ at the Semantic Publishing Challenge 2015 competition.

Semantic Vector-based Recommendations of Scholarly Documents. Since the emergence of academic digital libraries, like CiteSeer in the late nineties, recommendation of scholarly literature has been a thriving research topic. While existing approaches strive to recommend articles based on their bibliographical metadata or citation networks, in this dissertation, we proposed a new methodology for providing users with documents, whose *content* matches a user’s research interests. We investigated how the conventional information retrieval techniques, like the Vector Space Model,

can be adorned with semantic entities found within a document. Our results showed that by looking at the topics (i.e., named entities) in a document, the agent can outperform a conventional retrieval model that uses the surface forms of all the words mentioned in a document’s full-text. We also demonstrated that the rhetorical zones of a document can be used to provide users with a condensed summary of a document before recommending a user to read its full-text, thereby, effectively reducing her information overload.

Formulating Personalized Scholarly Services. Our work is the first to propose a formal model for a description of personal research agents, their tasks and workflow in RDF. Although the implementation of a complete human-computer interface for our agent was outside of the scope of this dissertation, as part of our contribution, we enumerated a set of scholarly services, derived from the requirements we gathered from surveys of researchers’ habit in working with scientific literature. We populated the agent’s knowledge base with a set of open access computer science publications and implemented the agent’s services as SPARQL queries over the knowledge base.


Table 29 revisits our research goals, the resulting contributions and the dissemination of their respective results.

9.2 Future Work

As an interdisciplinary research, our work in this dissertation can be extended in multiple directions:

In detecting the topics of a scholarly document, we used a named entity recognition tool (DBpedia Spotlight) that could ground the topics’ surface forms to a shallow, cross-domain ontology created based on an encyclopedia – Wikipedia. Since Spotlight was trained on general world knowledge, it demonstrated to fall short in linking highly specific topics in the computer science domain with their correct sense in the ontology. As a future work, one could look into the construction of a rich ontology of computer science domain entities, similar to the massive ontologies available in the bioinformatics domain, or perhaps develop an improved NER methodology for processing scientific articles.

As we explained earlier in this dissertation, several researchers have proposed various rhetorical entities and possible argumentation moves in scholarly writing. In this work, we limited the automatic capturing of rhetorical entities to Claim and Contribution sentences, as they were adequate to realize our agent’s example services. We see two possible extension points here: (i) we could investigate adding further rhetorical entity classes, e.g., Methods, to our schema and text mining solutions, and (ii) use the existing solution to generate labeled data and bootstrap large-scale, robust extraction of claims and contributions with machine- and deep-learning techniques.




Search

[Special page](#)

Navigation Main page Recent changes New Publication Tools Special pages	<h2 style="background-color: #d1c4e9; padding: 5px;">New Publication</h2> <p>Fill in the required information to create a new publication entry in the system.</p> <p style="color: red;">All fields are required</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>New Publication</p> <p>Enter a URL address where the publication can be fetched from.</p> <p>Article URL: <input type="text" value="http://www.semanticsoftware.info/system/files/mobiwist3_android.pdf"/></p> <p>Enter a desired name for the wiki page being created with the analysis results.</p> <p>Page Name: <input type="text" value="Sateli-MOBIIWIS2013"/></p> </div> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>Available Assistants</p> <p>Choose the services you would like to run on the paper. At least one service must be chosen.</p> <p><input checked="" type="checkbox"/> Claims and Contribution Extraction (Extracts claims and contributions from scholarly publications.)</p> <p><input checked="" type="checkbox"/> Readability Metrics (Measures the readability of a given block of text.)</p> <p><input type="checkbox"/> Automatic Indexer (Creates a back-of-the-book style index from noun phrases.)</p> </div> <div style="text-align: center; margin-top: 10px;"> <input type="button" value="Analyze"/> <input type="button" value="Refresh Services"/> </div> <p style="text-align: center;">⚙ Invoking selected services...</p>
--	--

(A) The wiki page to upload a document and execute various scholarly services



Search

[Refresh](#) [Watch](#) [Protect](#) [Move](#) [Delete](#) [History](#) [Edit](#) [Edit with form](#) [Discussion](#) [Page](#)

Navigation Main page Recent changes New Publication Tools What links here Related changes Special pages Printable version Permanent link Browse properties	<h2 style="background-color: #d1c4e9; padding: 5px;">Sateli-MOBIIWIS2013</h2> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> Publication Infosheet Claims and Contributions Readability Metrics Reviews [edit] </div> <div style="margin-top: 10px;"> <p>Claims:</p> <ul style="list-style-type: none"> Our approach introduces a novel Human-AI collaboration pattern that can be leveraged to aid mobile users with information-intensive tasks across various domains, such as health care, law, engineering, e-learning, e-business, among others. <p>Contributions:</p> <ul style="list-style-type: none"> We present a novel way of integrating NLP into Android applications. We demonstrate the applicability of these ideas with our open source Android library, based on the Semantic Assistants framework, and a prototype application 'iForgotWho' that detects names, numbers and organizations in user content and automatically enters them into the contact book. In this paper, we present the first open source NLP library for the Android platform that allows various applications to benefit from arbitrary NLP services through a comprehensive, service-oriented architecture. In what follows, we present a number of standard NLP tasks, with a focus on those relevant for mobile applications. As a part of our contribution and in order to demonstrate a general-purpose app offering arbitrary NLP services to Android mobile users, we have implemented an Android app, called the Semantic Assistants App, that offers a unique user interface to inquire and invoke NLP services on a user-provided content. To better demonstrate this use case, we implemented the iForgotWho (iFW) Android app and used its NLP capability on an example email message. Dates, locations and people can be automatically detected using named entity recognition and integrated in the creation of new events in a user's agenda and entries in the contact book as we demonstrated with the iFW app. </div>
--	--

(B) The detected Claim and Contribution sentences

Figure 43: The Zeeva wiki user interface

An impending threat to knowledge-based systems using graph structures for information modeling are inefficient, slow query times compared to relational databases. If our agent’s knowledge base is expected to contain a web-scale snapshot of a scholarly landscape, a detailed study of optimization techniques over graph-based query languages, such as SPARQL, is crucial.

Looking back at the semantic web stack (see Figure 7), our knowledge base still lacks the necessary formalization to enable the agent to conduct automatic semantic inferencing. A rather interesting opportunity here is to augment our semantic models with logical constructs, like axioms, or through redefining them in more rigid vocabularies, like the Web Ontology Language (OWL), thereby, facilitating automated reasoning over the knowledge base content.

Another interesting research direction to pursue in the future is investigating human-agent interaction patterns suitable for the scientific domain. We looked into the implementation of a Wiki-based system to integrate the agent’s services within a collaborative environment and implemented the Zeeva wiki [17, 18, 19] shown in Figure 43. However, we believe that an in-depth analysis of a user friendly environment, through which a personal research agent can proactively offer context- or task-sensitive tasks to its users, would greatly facilitate and expedite the adoption of such intelligent entities in the daily tasks of researchers.

Finally, the scholarly services mentioned in this dissertation by no means form an exhaustive list. A multitude of novel services can be envisaged by populating the agent’s knowledge base with additional artifacts, such as datasets or source code related to a published work, and devise new use cases that no other existing tools can offer to scholarly end-users, and help them to stay “on top of science”.

Bibliography

- [ABK⁺07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *Proceedings of the 6th International The Semantic Web and 2nd Conference on Asian Semantic Web Conference, ISWC'07 + ASWC'07*, pages 722–735, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-540-76298-0_52.
- [AKM⁺09] Teresa Attwood, Douglas Kell, Philip McDermott, James Marsh, Steve Pettifer, and David Thorne. Calling International Rescue: knowledge lost in literature and data landslide! *Biochemical Journal*, 424:317–333, 2009. <http://dx.doi.org/10.1042/BJ20091474>.
- [AL03] Laurence Anthony and George V. Lashkia. Mover: A machine learning tool to assist in the reading and writing of technical papers. *IEEE Transactions on professional communication*, 46(3):185–193, 2003. <http://dx.doi.org/10.1109/TPC.2003.816789>.
- [AMB⁺17] Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch, and Richard Cyganiak. Linking Open Data cloud diagram 2017. <http://lod-cloud.net/>, 2017.
- [Ant99] Laurence Anthony. Writing research article introductions in software engineering: How accurate is a standard model? *IEEE Transactions on Professional Communication*, 42(1):38–46, 1999. <http://dx.doi.org/10.1109/47.749366>.
- [BB10a] Georgeta Bordea and Paul Buitelaar. DERIUNLP: A Context Based Approach to Automatic Keyphrase Extraction. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval '10*, pages 146–149, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

- [BB10b] Georgetas Bordea and Paul Buitelaar. Expertise mining. In *Proceedings of the 21st National Conference on Artificial Intelligence and Cognitive Science*, Galway, Ireland, 2010.
- [BC76] Abraham Bookstein and William Cooper. A General Mathematical Model for Information Retrieval Systems. *The Library Quarterly: Information, Community, Policy*, 46(2):153–167, 1976. <http://dx.doi.org/10.1086/620501>.
- [BCD⁺12] Phil E. Bourne, Tim Clark, Robert Dale, Anita de Waard, Ivan Herman, Eduard Hovy, and David Shotton. Force11 Manifesto: Improving future research communication and e-scholarship. *White paper*, 2012. Retrieved online at: http://force11.org/white_paper.
- [BE08] Paul Buitelaar and Thomas Eigner. Topic Extraction from Scientific Literature for Competency Management. In *Proceedings of the 7th International Semantic Web Conference (ISWC'08)*, pages 25–66, Karlsruhe, Germany, 2008.
- [BKBP12] Georgeta Bordea, Sabrina Kirrane, Paul Buitelaar, and Bianca Pereira. Expertise Mining for Enterprise Content Management. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 3495–3498, Istanbul, Turkey, May 2012.
- [BL06] Tim Berners-Lee. Linked Data. <https://www.w3.org/DesignIssues/LinkedData.html>, 2006.
- [BL09] David M. Blei and John D. Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10(71):34, 2009.
- [BLG00] Kurt D. Bollacker, Steve Lawrence, and C. Lee Giles. Discovering relevant scientific literature on the web. *Intelligent Systems and their Applications*, 15(2):42–47, 2000.
- [BLH01] Tim Berners-Lee and James Hendler. Publishing on the semantic web. *Nature*, 410(6832):1023–1024, 2001.
- [BM07] Peter Brusilovsky and Eva Millán. User Models for Adaptive Hypermedia and Adaptive Educational Systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 3–53. Springer Berlin Heidelberg, 2007. http://dx.doi.org/10.1007/978-3-540-72079-9_1.
- [BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning research*, 3(2003):993–1022, 2003.

- [BOB82] Nicholas J. Belkin, Robert N. Oddy, and Helen M. Brooks. ASK for information retrieval: Part I. Background and theory. *Journal of documentation*, 38(2):61–71, 1982.
- [BV04] Chris Buckley and Ellen M. Voorhees. Retrieval Evaluation with Incomplete Information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 25–32, New York, NY, USA, 2004. ACM. <http://dx.doi.org/10.1145/1008992.1009000>.
- [Cao15] Longbing Cao. Agent Service-Oriented Architectural Design. In *Metasynthetic Computing and Engineering of Complex Systems*, pages 195–219. Springer London, 2015. http://dx.doi.org/10.1007/978-1-4471-6551-4_10.
- [CC99] Olga Caprotti and David Carlisle. OpenMath and MathML: Semantic Markup for Mathematics. *Crossroads*, 6(2):11–14, November 1999. <http://doi.acm.org/10.1145/333104.333110>.
- [CCMR⁺06] Ann Copestake, Peter Corbett, Peter Murray-Rust, CJ Rupp, Advait Siddharthan, Simone Teufel, and Ben Waldron. An architecture for language processing for scientific texts. In *Proceedings of the UK e-Science All Hands Meeting*, Nottingham, UK, 2006.
- [CMB⁺11] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. *Text Processing with GATE (Version 6)*. University of Sheffield, 2011.
- [CMBT02] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, PA, USA, 2002.
- [CMT00] Hamish Cunningham, Diana Maynard, and Valentin Tablan. JAPE: A Java Annotation Patterns Engine. Technical report, Department of Computer Science, University of Sheffield, 2000.
- [COGC⁺11] Paolo Ciccarese, Marco Ocana, Leyla Jael Garcia-Castro, Sudeshna Das, and Tim Clark. An open annotation ontology for science on web 3.0. *Journal of Biomedical Semantics*, 2(Suppl 2):S4, 2011. <http://dx.doi.org/10.1186/2041-1480-2-S2-S4>.

- [Coh04] Mike Cohn. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- [CP16] Elena Cotos and Nick Pendar. Discourse classification into rhetorical functions for AWE feedback. *CALICO Journal*, 33(1):92, 2016. <http://dx.doi.org/10.1558/cj.v33i1.27047>.
- [CPV13] Alexandru Constantin, Steve Pettifer, and Andrei Voronkov. PDFX: Fully-automated PDF-to-XML Conversion of Scientific Literature. In *Proceedings of the 2013 ACM Symposium on Document Engineering*, DocEng '13, pages 177–180, New York, NY, USA, 2013. ACM. <http://doi.acm.org/10.1145/2494266.2494271>.
- [CSRH13] Keith Cortis, Simon Scerri, Ismael Rivera, and Siegfried Handschuh. An Ontology-Based Technique for Online Profile Resolution. In *Social Informatics*, volume 8238 of *Lecture Notes in Computer Science*, pages 284–298. Springer International Publishing, 2013. http://dx.doi.org/10.1007/978-3-319-03260-3_25.
- [CWL14] Richard Cyganiak, David Wood, and Markus Lanthaler. Resource description framework (RDF): Concepts and abstract syntax, 2014. W3C Recommendation, <http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>.
- [CYY14] Ehtzaz Chaudhry, Xiaosong Yang, and Lihua You. Promoting Scientific Creativity by Utilising Web-based Research Objects: Deliverable No. 2.1 User requirement and use case report, 2014. <http://drinventor.eu/D9.2.1%20STYearDisseminationExploitationReport.pdf>.
- [DIPV09] Angelo Di Iorio, Silvio Peroni, and Fabio Vitali. Towards markup support for full GODDAGs and beyond: the EARMARK approach. In *Proceedings of Balisage: The Markup Conference*, Montréal, QC, Canada, 2009.
- [DJHM13] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, Graz, Austria, 2013.
- [DL82] Randall Davis and Douglas B. Lenat. *Knowledge-based Systems in Artificial Intelligence*. McGraw-Hill, 1982.
- [DM06] Fotis Draganidis and Gregoris Mentzas. Competency based management: a review of systems and approaches. *Information Management and Computer Security*, 14(1):51–64, 2006. <http://dx.doi.org/10.1108/09685220610648373>.

- [dRF17] Hélène de Ribaupierre and Gilles Falquet. Extracting discourse elements and annotating scientific documents using the SciAnnotDoc model: a use case in gender documents. *International Journal on Digital Libraries*, 2017. <http://dx.doi.org/10.1007/s00799-017-0227-5>.
- [Dub04] David Dubin. The Most Influential Paper Gerard Salton Never Wrote. *Graduate School of Library and Information Science. University of Illinois at Urbana-Champaign*, 2004.
- [dWT06] Anita de Waard and Gerard Tel. The ABCDE Format Enabling Semantic Conference Proceedings. In *The 1st Workshop: SemWiki 2006 – From Wiki to Semantics*, Budva, Montenegro, 2006.
- [FG97] Stan Franklin and Art Graesser. Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. In *Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, ECAI '96, pages 21–35, London, UK, 1997. Springer-Verlag. <http://dx.doi.org/10.1007/BFb0013570>.
- [FGC⁺08] Umer Farooq, Craig H. Ganoë, John M. Carroll, Isaac G. Council, and C. Lee Giles. Design and evaluation of awareness mechanisms in CiteSeer. *Information Processing & Management*, 44(2):596–612, 2008. <http://dx.doi.org/10.1016/j.ipm.2007.05.009>.
- [FPT⁺04] Valéria D. Feltrim, Jorge M. Pelizzoni, Simone Teufel, Maria das Graças Volpe Nunes, and Sandra M. Aluísio. Applying argumentative zoning in an automatic critiquer of academic writing. In *Brazilian Symposium on Artificial Intelligence*, pages 214–223. Springer, 2004.
- [GB10] Lisa Goddard and Gillian Byrne. The strongest link: Libraries and linked data. *D-Lib Magazine*, 2010. <http://dx.doi.org/10.1045/november2010-byrne>.
- [GBH09] Bela Gipp, Jöran Beel, and Christian Hentschel. Scienstein: A research paper recommender system. In *International Conference on Emerging Trends in Computing (ICETiC)*, pages 309–315, Virudhunagar, India, 2009.
- [GHMD07] Tudor Groza, Siegfried Handschuh, Knud Möller, and Stefan Decker. SALT – Semantically Annotated LaTeX for Scientific Publications. In *Proceedings of the 4th European conference on The Semantic Web: Research and Applications*, pages 518–532. Springer, Innsbruck, Austria, 2007. http://dx.doi.org/10.1007/978-3-540-72667-8_37.
- [GHMD08] Tudor Groza, Siegfried Handschuh, Knud Möller, and Stefan Decker. KonneX^{SALT}: First Steps Towards a Semantic Claim Federation Infrastructure. In Sean Bechhofer,

- Manfred Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, editors, *The Semantic Web: Research and Applications*, volume 5021 of *Lecture Notes in Computer Science*, pages 80–94. Springer Berlin Heidelberg, 2008. http://dx.doi.org/10.1007/978-3-540-68234-9_9.
- [HB11] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis lectures on the semantic web: theory and technology. Morgan & Claypool Publishers, 2011.
- [Hen01] James Hendler. Agents and the Semantic Web. *IEEE Intelligent systems*, 16(2):30–37, 2001. <http://dx.doi.org/10.1109/5254.920597>.
- [Hep00] Mark Hepple. Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 278–277, Hong Kong, 2000. Association for Computational Linguistics.
- [HP10] Konstantin Hyppönen and Vivian Michael Paganuzzi. Computer science research articles: the locations of different section types, and a proposal for standardization in the structure. *Scientometrics*, 84(1):199–220, July 2010. <http://dx.doi.org/10.1007/s11192-009-0089-8>.
- [HXC04] HR-XML-Consortium. Competencies (Measurable Characteristics). http://www.ec.tuwien.ac.at/~dorn/Courses/KM/Resources/hrxml/HR-XML-2_3/CP0/Competencies.html, 2004.
- [Hyl98] Ken Hyland. Persuasion and context: The pragmatics of academic metadiscourse. *Journal of pragmatics*, 30(4):437–455, 1998. [http://dx.doi.org/10.1016/S0378-2166\(98\)00009-5](http://dx.doi.org/10.1016/S0378-2166(98)00009-5).
- [Kan97] Noriko Kando. Text-Level Structure of Research Papers: Implications for Text-Based Information Processing Systems. In *BCS-IRSG Annual Colloquium on IR Research*, Aberdeen, UK, 1997.
- [KB15] Bianca Kramer and Jeroen Bosman. 101 innovations in scholarly communication: The changing research workflow. <https://doi.org/10.6084/m9.figshare.1286826.v1>, 2015. Poster.
- [Koh08] Michael Kohlhase. Using LaTeX as a Semantic Markup Format. *Mathematics in Computer Science*, 2(2):279–304, 2008. <http://dx.doi.org/10.1007/s11786-008-0055-5>.

- [Kuh15] Tobias Kuhn. Science Bots: A Model for the Future of Scientific Computation? In *WWW 2015 Companion Proceedings*, Florence, Italy, 2015. ACM. <http://dx.doi.org/10.1145/2740908.2742014>.
- [Lav08] Paul J. Lavrakas. *Encyclopedia of survey research methods*. Sage Publications, 2008.
- [Lid91] Elizabeth DuRoss Liddy. The discourse-level structure of empirical abstracts: An exploratory study. *Information Processing & Management*, 27(1):55–81, 1991. [http://dx.doi.org/10.1016/0306-4573\(91\)90031-G](http://dx.doi.org/10.1016/0306-4573(91)90031-G).
- [LPDB10] Julie Letierce, Alexandre Passant, Stefan Decker, and John G. Breslin. Understanding how Twitter is used to spread scientific messages. In *Web Science Conference*, 2010.
- [LPF⁺12] Yongming Luo, François Picalausa, George H. L. Fletcher, Jan Hidders, and Stijn Vansummen. Storing and Indexing Massive RDF Datasets. In Roberto De Virgilio, Francesco Guerra, and Yannis Velegrakis, editors, *Semantic Search over the Web*, pages 31–60. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. http://dx.doi.org/10.1007/978-3-642-25008-8_2.
- [LS08] Maria Liakata and Larisa Soldatova. Guidelines for the annotation of general scientific concepts. *Aberystwyth University, JISC Project Report*, 2008. http://repository.jisc.ac.uk/88/1/annot_guidelines.pdf.
- [LSD⁺12] Maria Liakata, Shyamasree Saha, Simon Dobnik, Colin R. Batchelor, and Dietrich Rebholz-Schuhmann. Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics*, 28(7):991–1000, 2012. <http://dx.doi.org/10.1093/bioinformatics/bts071>.
- [LTSB10] Maria Liakata, Simone Teufel, Advaith Siddharthan, and Colin R. Batchelor. Corpora for the Conceptualisation and Zoning of Scientific Papers. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'2010)*, pages 2054–2061, Valletta, Malta, 2010. ELDA.
- [Mae95] Pattie Maes. Artificial Life Meets Entertainment: Lifelike Autonomous Agents. *Commun. ACM*, 38(11):108–114, November 1995. <http://dx.doi.org/10.1145/219717.219808>.
- [Mar99] Daniel Marcu. A decision-based approach to rhetorical parsing. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 365–372, College Park, Maryland, 1999. Association for Computational Linguistics.

- [Min12] Gary Miner. *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press, 2012.
- [MM11] Raquel Mochales and Marie-Francine Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.
- [MMS93] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.
- [MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.
- [MSDR04] Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):54–88, 2004. <http://dx.doi.org/10.1145/963770.963773>.
- [MYGHA13] Ashutosh Malhotra, Erfan Younesi, Harsha Gurulingappa, and Martin Hofmann-Apitius. ‘HypothesisFinder:’ A Strategy for the Detection of Speculative Statements in Scientific Text. *PLoS computational biology*, 9(7):e1003117, 2013.
- [NHA08] Amine Naak, Hicham Hage, and Esma Aïmeur. Papyres: A Research Paper Management System. In *E-Commerce Technology and the 5th IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, pages 201–208, Washington, DC, USA, July 2008. <http://dx.doi.org/10.1109/CECandEEE.2008.132>.
- [NHA09] Amine Naak, Hicham Hage, and Esma Aïmeur. A Multi-criteria Collaborative Filtering Approach for Research Paper Recommendation in Papyres. In Gilbert Babin, Peter Kropf, and Michael Weiss, editors, *E-Technologies: Innovation in an Open World*, volume 26 of *Lecture Notes in Business Information Processing*, pages 25–39. Springer Berlin Heidelberg, 2009. http://dx.doi.org/10.1007/978-3-642-01187-0_3.
- [NN14] P. K. Ramachandran Nair and Vimala D. Nair. *Organization of a Research Paper: The IMRAD Format*, pages 13–25. Springer International Publishing, 2014. http://dx.doi.org/10.1007/978-3-319-03101-9_2.
- [NS07] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007. <http://dx.doi.org/10.1075/li.30.1.03nad>.

- [NS16] Chifumi Nishioka and Ansgar Scherp. Profiling vs. Time vs. Content: What Does Matter for Top-k Publication Recommendation Based on Twitter Profiles? In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, JCDL '16, pages 171–180, New York, NY, USA, 2016. ACM. <http://dx.doi.org/10.1007/10.1145/2910896.2910898>.
- [OPO⁺14] Diarmuid P. O'Donoghue, James Power, Sian O'Briain, Feng Dong, Aidan Mooney, Donny Hurley, Yalemisew Abgaz, and Charles Markham. Can a Computationally Creative System Create Itself? Creative Artefacts and Creative Processes. In *Proceedings of the 5th International Conference on Computational Creativity*, Ljubljana, Slovenia, 2014. Jožef Stefan Institute.
- [PC08] Nick Pendar and Elena Cotos. Automatic Identification of Discourse Moves in Scientific Article Introductions. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, EANL '08, pages 62–70, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.
- [Per12] Silvio Peroni. *Semantic Publishing: issues, solutions and new trends in scholarly publishing within the Semantic Web era*. PhD Dissertation, University of Bologna, 2012. http://amsdottorato.unibo.it/4766/1/peroni_silvio_tesi.pdf.
- [RBC⁺07] Patrick Ruch, Celia Boyer, Christine Chichester, Imad Tbahrity, Antoine Geissbühler, Paul Fabry, Julien Gobeill, Violaine Pillet, Dietrich Rebholz-Schuhmann, Christian Lovis, et al. Using argumentation to extract key sentences from biomedical abstracts. *International journal of medical informatics*, 76(2):195–200, 2007. <http://dx.doi.org/10.1016/j.ijmedinf.2006.05.002>.
- [RCTW06] CJ Rupp, Ann Copestake, Simone Teufel, and Ben Waldron. Flexible interfaces in the application of language technology to an eScience corpus. In *Proceedings of the UK e-Science Programme All Hands Meeting*, Nottingham, UK, 2006.
- [Ril17] Jenn Riley. *Understanding Metadata: What is Metadata and What is it for?* National Information Standards Organization (NISO), 2017. <http://dx.doi.org/10.1080/01639374.2017.1358232>.
- [RPM⁺14] Yuan Ren, Artemis Parvizi, Chris Mellish, Jeff Z. Pan, Kees Van Deemter, and Robert Stevens. Towards competency question-driven ontology authoring. In *Extended Semantic Web Conference (ESWC)*, pages 752–767, Heraklion, Crete, Greece, 2014. Springer. http://dx.doi.org/10.1007/978-3-319-07443-6_50.

- [RS15] Francesco Ronzano and Horacio Saggion. Dr. Inventor Framework: Extracting Structured Information from Scientific Publications. In *Discovery Science*, pages 209–220. Springer, 2015. http://dx.doi.org/10.1007/978-3-319-24282-8_18.
- [RW94] Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241. Springer-Verlag New York, Inc., 1994. http://dx.doi.org/10.1007/978-1-4471-2099-5_24.
- [Sal89] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [SCSK06] Larisa N. Soldatova, Amanda Clare, Andrew Sparkes, and Ross D. King. An ontology for a Robot Scientist. *Bioinformatics*, 22(14):e464–e471, 2006. <http://dx.doi.org/10.1093/bioinformatics/btl207>.
- [Sho09] David Shotton. Semantic publishing: the coming revolution in scientific journal publishing. *Learned Publishing*, 22(2):85–94, 2009. <http://dx.doi.org/10.1087/2009202>.
- [Sho12] David Shotton. The five stars of online journal articles – a Framework for Article Evaluation. *D-Lib Magazine*, 18(1/2), 2012. <http://dx.doi.org/10.1045/january2012-shotton>.
- [SJWR00] Karen Spärck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information processing & management*, 36(6):809–840, 2000. [http://dx.doi.org/10.1016/S0306-4573\(00\)00016-9](http://dx.doi.org/10.1016/S0306-4573(00)00016-9).
- [SK10] Kazunari Sugiyama and Min-Yen Kan. Scholarly Paper Recommendation via User’s Recent Research Interests. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries, JCDL ’10*, pages 29–38, New York, NY, USA, 2010. ACM. <http://dx.doi.org/10.1145/1816123.1816129>.
- [SKW07] 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. <http://dx.doi.org/10.1145/1242572.1242667>.

- [SPKM09] David Shotton, Katie Portwin, Graham Klyne, and Alistair Miles. Adventures in semantic publishing: exemplar semantic enhancements of a research article. *PLoS Computational Biology*, 5(4):e1000361, 2009. <http://dx.doi.org/10.1371/journal.pcbi.1000361>.
- [Swa90] John Swales. *Genre analysis: English in academic and research settings*. Cambridge University Press, 1990.
- [SWY75] Gerard Salton, Andrew Wong, and Chungshu Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, November 1975. <http://dx.doi.org/10.1145/361219.361220>.
- [TCM99] Simone Teufel, Jean Carletta, and Marc Moens. An Annotation Scheme for Discourse-level Argumentation in Research Articles. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, pages 110–117, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics. <http://dx.doi.org/10.3115/977035.977051>.
- [Teo06] Tina Teodorescu. Competence Versus Competency: What is the difference? *Performance Improvement*, 45(10), 2006. <http://dx.doi.org/10.1002/pfi.4930451027>.
- [Teu99] Simone Teufel. *Argumentative zoning: Information extraction from scientific text*. PhD thesis, University of Edinburgh, 1999. <https://www.cl.cam.ac.uk/~sht25/thesis/t1.pdf>.
- [Teu06] Simone Teufel. Argumentative zoning for improved citation indexing. In *Computing Attitude and Affect in Text: Theory and Applications*, pages 159–169. Springer, 2006. http://dx.doi.org/10.1007/1-4020-4102-0_13.
- [Teu10] Simone Teufel. *The Structure of Scientific Articles - Applications to Citation Indexing and Summarization*. CSLI Studies in Computational Linguistics. University of Chicago Press, 2010.
- [TM02] Simone Teufel and Marc Moens. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445, 2002. <http://dx.doi.org/10.1162/089120102762671936>.
- [TSB09] Simone Teufel, Advait Siddharthan, and Colin R. Batchelor. Towards Discipline-independent Argumentative Zoning: Evidence from Chemistry and Computational Linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural*

Language Processing: Volume 3, EMNLP '09, pages 1493–1502, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

- [TYZZ10] Jie Tang, Limin Yao, Duo Zhang, and Jing Zhang. A Combination Approach to Web User Profiling. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(1):1–39, December 2010. <http://dx.doi.org/10.1145/1870096.1870098>.
- [UG96] Mike Uschold and Michael Gruninger. Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(2):93–136, 1996. <http://dx.doi.org/10.1017/S0269888900007797>.
- [WT88] Mann William and Sandra Thompson. Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8(3):243–281, 1988. <http://dx.doi.org/10.1515/text.1.1988.8.3.243>.
- [WZRH14] David Wood, Marsha Zaidman, Luke Ruth, and Michael Hausenblas. *Linked Data: Structured data on the Web*. Manning Publications Co., 2014.
- [Yan09] Yan, Ying and Wang, Chen and Zhou, Aoying and Qian, Weining and Ma, Li and Pan, Yue. Efficient indices using graph partitioning in RDF triple stores. In *The 25th International Conference on Data Engineering (ICDE'09)*, pages 1263–1266, Shanghai, China, 2009. IEEE. <http://dx.doi.org/10.1109/ICDE.2009.216>.
- [ZGG13] Leila Zemmouchi-Ghomari and Abdessamed Réda Ghomari. Translating natural language competency questions into SPARQL Queries: a case study. In *The 1st International Conference on Building and Exploring Web Based Environments*, pages 81–86, Seville, Spain, 2013.

Author's Publications

- [1] Fedor Bakalov, Marie-Jean Meurs, Birgitta König-Ries, Bahar Sateli, René Witte, Greg Butler, and Adrian Tsang. Empowering web portal users with personalized text mining services. In *NETTAB 2012 workshop focused on Integrated Bio-Search*, volume 18 (Supplement B), pages 81–83, Como, Italy, November 2012. EMBnet.journal. **Received “Best Poster Award” at NETTAB 2012.** <http://journal.embnet.org/index.php/embnetjournal/article/view/558>.
- [2] Fedor Bakalov, Marie-Jean Meurs, Birgitta König-Ries, Bahar Sateli, René Witte, Greg Butler, and Adrian Tsang. Personalized Semantic Assistance for the Curation of Biochemical Literature. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2012)*, Philadelphia, PA, USA, October 2012. IEEE. <http://dx.doi.org/10.1109/BIBM.2012.6392735>.
- [3] Fedor Bakalov, Marie-Jean Meurs, Birgitta König-Ries, Bahar Sateli, René Witte, Greg Butler, and Adrian Tsang. An Approach to Controlling User Models and Personalization Effects in Recommender Systems. In *International Conference on Intelligent User Interfaces (IUI '13)*, page 49–56, Santa Monica, CA, USA, March 2013. ACM. <http://dl.acm.org/authorize?6808714>.
- [4] Fedor Bakalov, Bahar Sateli, René Witte, Marie-Jean Meurs, and Birgitta König-Ries. Natural Language Processing for Semantic Assistance in Web Portals. In *IEEE International Conference on Semantic Computing (ICSC 2012)*, Palermo, Italy, September 2012. IEEE. <http://dx.doi.org/10.1109/ICSC.2012.38>.
- [5] Felicitas Löffler, Bahar Sateli, Birgitta König-Ries, and René Witte. Semantic Content Processing in Web Portals. In *4th Canadian Semantic Web Symposium (CSWS 2013)*, volume 1054 of *CEUR Workshop Proceedings*, page 50–51, Montréal, QC, Canada, July 2013. CEUR-WS.org. <http://ceur-ws.org/Vol-1054/paper-12.pdf>.

- [6] Felicitas Löffler, Bahar Sateli, René Witte, and Birgitta König-Ries. Towards Semantic Recommendation of Biodiversity Datasets based on Linked Open Data. In *26. GI-Workshop Grundlagen von Datenbanken*, Ritten, Italy, October 2014. http://ceur-ws.org/Vol-1313/paper_12.pdf.
- [7] Bahar Sateli. A General Architecture to Enhance Wiki Systems with Natural Language Processing Techniques. Master's thesis, Concordia University, Montreal, April 2012. <http://spectrum.library.concordia.ca/974058/>.
- [8] Bahar Sateli. Semantic Management of Scholarly Literature: A Wiki-based Approach. In Marina Sokolova and Peter van Beek, editors, *The 27th Canadian Conference on Artificial Intelligence (Canadian AI 2014)*, volume LNCS 8436 of *Advances in Artificial Intelligence*, page 387–392, Montréal, Canada, April 2014. Springer. http://dx.doi.org/10.1007/978-3-319-06483-3_43.
- [9] Bahar Sateli, Gina Cook, and René Witte. Smarter Mobile Apps through Integrated Natural Language Processing Services. In Florian Daniel, George A. Papadopoulos, and Philippe Thiran, editors, *The 10th International Conference on Mobile Web Information Systems (MobiWIS 2013)*, volume 8093 of *Lecture Notes in Computer Science*, page 187–202, Paphos, Cyprus, August 2013. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-40276-0_15.
- [10] Bahar Sateli, Felicitas Löffler, Birgitta König-Ries, and René Witte. Semantic User Profiles: Learning Scholars' Competences by Analyzing their Publications. In *Semantics, Analytics, Visualisation: Enhancing Scholarly Data (SAVE-SD 2016)*, Montréal, QC, Canada, April 2016. ACM. <http://cs.unibo.it/save-sd/2016/papers/pdf/sateli-savesd2016.pdf>.
- [11] Bahar Sateli, Felicitas Löffler, Birgitta König-Ries, and René Witte. ScholarLens: extracting competences from research publications for the automatic generation of semantic user profiles. *PeerJ Computer Science*, 3:e121, 2017. <https://peerj.com/articles/cs-121/>.
- [12] Bahar Sateli, Sebastien Luong, and René Witte. TagCurate: crowdsourcing the verification of biomedical annotations to mobile users. In *NETTAB 2013 workshop focused on Semantic, Social, and Mobile Applications for Bioinformatics and Biomedical Laboratories*, volume 19 Suppl. B, page 24–26, Venice Lido, Italy, October 2013. EMBnet.journal. <http://journal.embnet.org/index.php/embnetjournal/article/view/722>.
- [13] Bahar Sateli, Marie-Jean Meurs, Greg Butler, Justin Powlowski, Adrian Tsang, and René Witte. IntelliGenWiki: An Intelligent Semantic Wiki for Life Sciences. In *NETTAB*

- 2012 workshop focused on Integrated Bio-Search*, volume 18 (Supplement B), page 50–52, Como, Italy, November 2012. EMBnet.journal. <http://journal.embnet.org/index.php/embnetjournal/article/view/547>.
- [14] Bahar Sateli, Caitlin Murphy, René Witte, Marie-Jean Meurs, and Adrian Tsang. Text Mining Assistants in Wikis for Biocuration. In *5th International Biocuration Conference*, page 126, Washington DC, USA, April 2012. International Society for Biocuration. <http://www.semanticsoftware.info/system/files/biocuration2012poster.png>.
 - [15] Bahar Sateli and René Witte. Natural Language Processing for MediaWiki: The Semantic Assistants Approach. In *The 8th International Symposium on Wikis and Open Collaboration (WikiSym 2012)*, Linz, Austria, August 2012. ACM. <http://www.opensym.org/ws2012/p20wikisym2012.pdf>.
 - [16] Bahar Sateli and René Witte. Supporting Wiki Users with Natural Language Processing. In *The 8th International Symposium on Wikis and Open Collaboration (WikiSym 2012)*, Linz, Austria, August 2012. ACM. <http://dx.doi.org/10.1145/2462932.2462976>.
 - [17] Bahar Sateli and René Witte. Collaborative Semantic Management and Automated Analysis of Scientific Literature. In *The 11th Extended Semantic Web Conference (ESWC 2014)*, Anissaras, Crete, Greece, May 2014. http://dx.doi.org/10.1007/978-3-319-11955-7_73.
 - [18] Bahar Sateli and René Witte. Semantic Management of Scholarly Literature: A Wiki-based Approach. In *The 9th Semantic MediaWiki Conference (SMWCon Spring 2014)*, Montréal, Canada, May 2014. <http://www.semanticsoftware.info/system/files/zeeva-smwcon14-talk.pdf>.
 - [19] Bahar Sateli and René Witte. Supporting Researchers with a Semantic Literature Management Wiki. In Alexander García Castro, Christoph Lange, Phillip W. Lord, and Robert Stevens, editors, *The 4th Workshop on Semantic Publishing (SePublica 2014)*, volume 1155 of *CEUR Workshop Proceedings*, Anissaras, Crete, Greece, May 2014. CEUR-WS.org. <http://ceur-ws.org/Vol-1155/paper-03.pdf>.
 - [20] Bahar Sateli and René Witte. Automatic Construction of a Semantic Knowledge Base from CEUR Workshop Proceedings. In *The 12th Extended Semantic Web Conference (The Semantic Publishing Challenge 2015)*, volume 548 of *Semantic Web Evaluation Challenges: SemWebEval 2015 at ESWC 2015, Revised Selected Papers*, page 129–141, Portoroz, Slovenia, June 2015. Springer. **Winner of the “Most Innovative Approach” Award**, http://dx.doi.org/10.1007/978-3-319-25518-7_11.

- [21] Bahar Sateli and René Witte. Semantic representation of scientific literature: bringing claims, contributions and named entities onto the Linked Open Data cloud. *PeerJ Computer Science*, 1(e37), 2015. <https://peerj.com/articles/cs-37/>.
- [22] Bahar Sateli and René Witte. What’s in this paper? Combining Rhetorical Entities with Linked Open Data for Semantic Literature Querying. In *Semantics, Analytics, Visualisation: Enhancing Scholarly Data (SAVE-SD 2015)*, page 1023–1028, Florence, Italy, May 2015. ACM. **Received “Best Paper Award”** at SAVE-SD 2015. <http://www.www2015.it/documents/proceedings/companion/p1023.pdf>.
- [23] Bahar Sateli and René Witte. An Automatic Workflow for Formalization of Scholarly Articles’ Structural and Semantic Elements. In Harald Sack, Stefan Dietze, Anna Tordai, and Christoph Lange, editors, *The 13th Extended Semantic Web Conference (The Semantic Publishing Challenge 2016)*, volume 641 of *Third SemWebEval Challenge at ESWC 2016, Revised Selected Papers*, page 309–320, Heraklion, Crete, Greece, June 2016. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-46565-4_24.
- [24] Bahar Sateli and René Witte. From Papers to Triples: An Open Source Workflow for Semantic Publishing Experiments. In Alejandra González-Beltrán, Francesco Osborne, and Silvio Peroni, editors, *Semantics, Analytics, Visualisation: Enhancing Scholarly Data (SAVE-SD 2016)*, page 39–44, Montréal, QC, Canada, April 2016. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-53637-8_5.
- [25] Bahar Sateli and René Witte. Personal Research Agents on the Web of Linked Open Data. In Jorge Gracia, Francis Bond, John P. McCrae, Paul Buitelaar, and Christian Chiarcos, editors, *Language, Data and Knowledge 2017 (LDK 2017)*, volume 10318 of *Lecture Notes in Computer Science (LNCS)*, pages 10–25, Galway, Ireland, May 2017. Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-59888-8_2.
- [26] Bahar Sateli and René Witte. Personal Research Assistants for Young Researchers. In *Writing Analytics Literacy – Bridging from Research to Practice (LAK 17)*, co-located with the 7th International Learning Analytics & Knowledge Conference, Vancouver, BC, Canada, March 2017. http://www.semanticsoftware.info/system/files/lak17_abstract.pdf.
- [27] René Witte and Bahar Sateli. Adding Natural Language Processing Support to your (Semantic) MediaWiki. In *The 9th Semantic MediaWiki Conference (SMWCon Spring 2014)*, Montreal, Canada, May 2014. Tutorial. https://semantic-mediawiki.org/wiki/SMWCon_Spring_2014/Wiki-NLP_Tutorial.

- [28] René Witte and Bahar Sateli. Combining Off-the-shelf Grammar and Spelling Tools for the Automatic Evaluation of Scientific Writing (AESW) Shared Task 2016. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@NAACL-HLT 2016, June 16, 2016, San Diego, California, USA*, pages 252–255, 2016. <http://www.aclweb.org/anthology/W16-0529>.
- [29] René Witte and Bahar Sateli. The LODEXporter: Flexible Generation of Linked Open Data Triples from NLP Frameworks for Automatic Knowledge Base Construction. In *International Conference on Language Resources and Evaluation (LREC)*, Miyazaki, Japan, May 2018. (In Press).

Appendix A

Supplementary Materials

The following online resources contain supplementary materials of this dissertation:

Main Repository. The datasets, results, scripts and populated KBs used in our work are available in a dedicated Github repository at https://github.com/baharsateli/Dissertation_Supplementary_Materials.

Semantic Publishing Challenge. The tools used in the Semantic Publishing Challenge 2015 and 2016, as well as the populated knowledge base and relevant queries:

- SemPub 2015, <http://www.semanticsoftware.info/sempub-challenge-2015>
- SemPub 2016, <http://www.semanticsoftware.info/sempub-challenge-2016>

Rhetector. Our text mining pipeline used to extract rhetorical entities of scholarly literature:

- Webpage, <http://www.semanticsoftware.info/rhetector>
- Source code, <https://github.com/SemanticSoftwareLab/TextMining-Rhetector>

LODtagger. Our text mining pipeline used to extract named entities in scholarly documents, using an external NER tool (e.g., DBpedia Spotlight):

- Webpage, <http://www.semanticsoftware.info/lodtagger>
- Source code, <https://github.com/SemanticSoftwareLab/TextMining-LODtagger>

LODeXporter. Our GATE component for transforming NLP annotations to RDF triples:

- Webpage, <http://www.semanticsoftware.info/lodexporter>
- Source code, <https://github.com/SemanticSoftwareLab/TextMining-LODeXporter>

ScholarLens. Our scholarly user profiling pipeline:

- Webpage, <http://www.semanticsoftware.info/semantic-user-profiling-peerj-2016-supplements>
- Source code, <https://github.com/SemanticSoftwareLab/ScholarLens>

Zeeva. A collaborative, wiki-based system for semantic management of scientific literature:

- Webpage, <http://www.semanticsoftware.info/zeeva>
- Underlying architecture, <http://www.semanticsoftware.info/semantic-assistants-wiki-nlp>

Appendix B

ANNIE Part-of-Speech Tagset

The following table provides the part-of-speech symbols used in GATE's ANNIE pipeline, in particular the POS tagger processing resource and JAPE rules. We limited the table to those mentioned in this manuscript. The complete tagset can be found online.¹

POS Tag	Description
DT	determiner: Articles including 'a', 'an', 'every', 'no', 'the', 'another', 'any', 'some', 'those'.
IN	preposition or subordinating conjunction
JJ	adjective: Hyphenated compounds that are used as modifiers; e.g., 'happy-go-lucky'.
JJR	adjective - comparative: Adjectives with the comparative ending '-er' and a comparative meaning. Sometimes 'more' and 'less'.
JJS	adjective - superlative: Adjectives with the superlative ending '-est' (and 'worst'). Sometimes 'most' and 'least'.
JJSS	probably a variant of JJS
NN	noun - singular or mass
NNP	proper noun - singular: All words in names usually are capitalized but titles might not be.
NNPS	proper noun - plural: All words in names usually are capitalized but titles might not be.
NNS	noun - plural
NP	proper noun - singular
NPS	proper noun - plural
POS	possessive ending: Nouns ending in 's' or apostrophe.
PP	personal pronoun
PRPR\$	probably possessive pronoun
PRP	probably possessive pronoun
PRP\$	probably possessive pronoun, such as 'my', 'your', 'his', 'his', 'its', 'one's', 'our', and 'their'.
RB	adverb: most words ending in '-ly'. Also 'quite', 'too', 'very', 'enough', 'indeed', 'not', '-n't', and 'never'.
RBR	adverb - comparative: adverbs ending with '-er' with a comparative meaning.

¹ANNIE POS Tags, <https://gate.ac.uk/sale/tao/splitap7.html#x39-786000G>

RBS	adverb - superlative
VBD	verb - past tense: includes conditional form of the verb 'to be'; e.g., 'If I were/VBD rich...'
VBG	verb - gerund or present participle
VCN	verb - past participle
VBP	verb - non-3rd person singular present
VB	verb - base form: subsumes imperatives, infinitives and subjunctives.
VBZ	verb - 3rd person singular present

Appendix C

Referenced Ontologies

The following table provides the ontology description and namespaces used in this dissertation's figures and listings, as well as their corresponding URIs.

Ontology	Namespace	URI	Description
BIBO	bibo	http://purl.org/ontology/bibo/	The Bibliographic Ontology
Competence	c	http://www.intelleo.eu/ontologies/competences/spec	IntelLEO Competence Management Ontology
DBpedia	dbpedia	http://dbpedia.org/resource/	DBpedia Ontology
DCTERMS	dcterms	http://purl.org/dc/terms/	Dublin Core Metadata Terms
DoCO	doco	http://purl.org/spar/doco/	Document Components Ontology
FOAF	foaf	http://xmlns.com/foaf/0.1/	Friend-of-a-Friend Ontology
GEO	gn	http://www.geonames.org/ontology#	The Geonames Ontology
Lifecycle	lifecycle	http://vocab.org/lifecycle/schema#	Lifecycle Ontology
MAPPING	map	http://lod.semanticsoftware.info/mapping/mapping#	LODeXporter Mapping Ontology
OA	oa	http://www.w3.org/ns/oa#	The Web Annotation Ontology
PRAV	prav	http://lod.semanticsoftware.info/prav/prav#	Our Personal Research Agent Vocabulary
PUBO	pubo	http://lod.semanticsoftware.info/pubo/pubo#	Our PUBlication Ontology
REL	rel	http://purl.org/vocab/relationship/	The Relationship Ontology
RDF	rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Resource Description Framework
RDFS	rdfs	http://www.w3.org/2000/01/rdf-schema#	RDF Schema
SRO	sro	http://salt.semanticauthoring.org/ontologies/sro#	SALT Rhetorical Ontology
UM	um	http://intelleo.eu/ontologies/user-model/ns/	IntelLEO User Model ontology
VCARD	vc	http://www.w3.org/2006/vcard/ns#&vCardOntology	The vCard Ontology
XSD	xsd	http://www.w3.org/2001/XMLSchema#	XML Schema Definition

Appendix D

Example Competency Questions

1. Which authors wrote `<ex:doc123>`?
2. Which affiliations is `<ex:author123>` associated with?
3. What is the title of `<ex:doc123>`?
4. What is the contact information of `<ex:author123>`?
5. Which documents have a Claim?
6. Which documents mention `<dbpedia:Software_prototyping>`?
7. Which documents have a contribution regarding `<dbpedia:Software_prototyping>`?
8. Which authors do research on `<dbpedia:Software_prototyping>`?
9. Which author published the highest number of articles in 2017?
10. Which affiliations published the highest number of articles between 2010 and 2017?
11. Which country published the highest number of articles in the ACL proceedings?
12. Which documents are similar to `<ex:doc123>`?
13. What does `<dbpedia:Software_prototyping>` mean?
14. Which documents should I read to learn about `<dbpedia:Software_prototyping>`?
15. Show me a summary of all documents that have ‘*linked data*’ in their title.
16. What European countries are doing research on `<dbpedia:Software_prototyping>`?
17. What top 5 topics were trending at the WWW conference in 2015?

Appendix E

Rhetorical Analysis Resources

Rhetector Lexical Resources

Type	Terms from the Action Lexicon
Change	manipulate, modify, combine, derive, filter, convert, adapt, discard, constrain, transform, alter, increase, decrease, refine, limit, divide, reduce, restrict, substitute, fabricate, shift, change, exclude, adjust, multiply, replace, augment, expand, extend, revise, tailor
Presentation	discuss, describe, illustrate, present, report, propose, show, exhibit, summarize, visualize, give, introduce, point out, put forward, sketch, outline, highlight
Research	investigate, calculate, recalculate, compute, examine, determine, apply, record, analyze, analyse, characterize, characterise, categorize, categorise, assess, identify, re-examine, verify, quantify, extrapolate, rationalize, rationalise, simulate, implement, inspect, classify, cluster, realize, realise, expect, conduct, specify, interpret, study, evaluate, predict, organize, organise, measure, process, observe, design, employ, define, build, collect, compose, construct, delineate, detect, estimate, maximise, maximize, minimize, minimise, reconfirm, select, test, aim, revisit, explore
Solution	solution, perform, demonstrate, answer, enhance, overcome, explain, automate, tackle, contribute, guarantee, develop, create, resolve, solve, obtain, address, design, enable, establish, improve, discover, accomplish, account for, achieve, apply to, alleviate, allow for, avoid, benefit, capture, clarify, circumvent, devise, elucidate, fix, gain, handle, implement, make progress, mend, manage, mitigate, model, offer, preserve, prove, reveal, succeed, warrant, suggest, conclude
Type	Terms from the Concept Lexicon
deictic phrases	article, case study, field study, finding, library, paper, position paper, presented work, project, prototype, research, research work, study, work
Entity	algorithm, approach, framework, methodology, platform, project, prototype, system
Verbs	argue, can, confirm, emphasize, find, formalize, obtain, show, succeed

Rhetector Rule-based Patterns

Deictics

RULE_{deictic₁}: PREPOSITION + DETERMINER + DICTIONARY_{deixis}

RULE_{deictic₂}: DETERMINER + UPPER-INITIAL TOKEN + DICTIONARY_{deixis}

RULE_{deictic₃}: POSSESSIVE PRONOUN + DICTIONARY_{deixis}

Metadiscourse

RULE_{metadiscourse₁}: DEICTIC + PUNCTUATION + DICTIONARY_{action}

RULE_{metadiscourse₂}: PERSONAL PRONOUN_(We|I) + ORDINAL + DICTIONARY_{action}

RULE_{metadiscourse₃}: DEICTIC + PUNCTUATION + POSSESSIVE PRONOUN_(Our|My) + ADVERB + VERB PHRASE? + DICTIONARY_{action}

Claims

RULE_{claim₁}: POSSESSIVE PRONOUN_(Our|My) + PROPER NOUN_{singular} + (DICTIONARY_{entity})? + ADVERB + VERB_{3rd person singular present} + ADJECTIVE_{comparative or superlative}

RULE_{claim₂}: POSSESSIVE PRONOUN_(Our|My) + PROPER NOUN_{singular} + DICTIONARY_{entity} + ADVERB + VERB_{past tense} + ADJECTIVE_{comparative or superlative}

RULE_{claim₃}: POSSESSIVE PRONOUN_(Our|My) + NOUN PHRASE + DICTIONARY_{entity} + ADVERB + VERB + DETERMINER + ADJECTIVE

RULE_{claim₃}: POSSESSIVE PRONOUN_(Our|My) + NOUN PHRASE + DICTIONARY_{entity} + VERB + ADJECTIVE

RULE_{claim₄}: POSSESSIVE PRONOUN_(Our|My) + NOUN PHRASE + DICTIONARY_{entity} + VERB + VERB_{past tense}

Appendix F

LODeXporter Mapping File

The following code shows the mapping file used to configure LODEXporter for our experiments, in the RDF Turtle¹ format.

```
1 @prefix map: <http://lod.semanticsoftware.info/mapping/mapping#> .
2 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
3 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
4 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
5 @prefix rel: <http://purl.org/vocab/relationship/> .
6 @prefix ex: <http://example.com/> .
7 @prefix cnt: <http://www.w3.org/2011/content#> .
8 @prefix oa: <http://www.w3.org/ns/oa#> .
9 @prefix gn: <http://www.geonames.org/ontology#> .
10 @prefix pubo: <http://lod.semanticsoftware.info/pubo/pubo#> .
11 @prefix sro: <http://salt.semanticauthoring.org/ontologies/sro#> .
12 @prefix doco: <http://purl.org/spar/doco/> .
13
14 <ex:RhetoricalEntityLinkedNamedEntityRelationMapping>
15   a <map:Mapping> ;
16   map:type <pubo:containsNE> ;
17   map:domain <ex:GATERhetoricalEntity> ;
18   map:range <ex:GATEDBpediaNE> ;
19   map:GATEattribute "contains" .
20
21 <ex:AuthorAffiliationRelationMapping>
22   a <map:Mapping> ;
23   map:type <rel:employedBy> ;
24   map:domain <ex:GATEAuthor> ;
25   map:range <ex:GATEAffiliation> ;
26   map:GATEattribute "employedBy" .
27
28 <ex:GATEContentMapping>
29   a <map:Mapping> ;
30   map:type <cnt:chars> ;
31   map:GATEattribute "content" .
```

¹RDF Turtle, <https://www.w3.org/TR/turtle/>

```

32
33 <ex:GATEStartOffsetMapping>
34   a <map:Mapping> ;
35   map:type <oa:start> ;
36   map:GATEattribute "startOffset" .
37
38 <ex:GATEEndOffsetMapping>
39   a <map:Mapping> ;
40   map:type <oa:end> ;
41   map:GATEattribute "endOffset" .
42
43 <ex:GATELODRefFeatureMapping>
44   a <map:Mapping> ;
45   map:type rdfs:isDefinedBy ;
46   map:GATEfeature "URI" .
47
48 <ex:GATEURIFeatureMapping>
49   a <map:Mapping> ;
50   map:type rdf:type ;
51   map:GATEfeature "URI1" .
52
53 <ex:GATELocatedInFeatureMapping>
54   a <map:Mapping> ;
55   map:type <gn:locatedIn> ;
56   map:GATEfeature "locatedIn" .
57
58 <ex:GATELocationURIFeatureMapping>
59   a <map:Mapping> ;
60   map:type rdfs:isDefinedBy ;
61   map:GATEfeature "locationURI" .
62
63 <ex:GATEFirstnameFeatureMapping>
64   a <map:Mapping> ;
65   map:type foaf:givenName ;
66   map:GATEfeature "firstname" .
67
68 <ex:GATELastnameFeatureMapping>
69   a <map:Mapping> ;
70   map:type foaf:familyName ;
71   map:GATEfeature "lastname" .
72
73 <ex:GATERhetoricalEntity>
74   a <map:Mapping> ;
75   map:hasMapping <ex:GATEContentMapping>,
76   <ex:GATEStartOffsetMapping>,
77   <ex:GATEEndOffsetMapping>,
78   <ex:GATEURIFeatureMapping> ;
79   map:baseURI <http://semanticsoftware.info/lodexporter/> ;
80   map:type <sro:RhetoricalElement> ;
81   map:GATEtype "RhetoricalEntity" .
82
83 <ex:GATEDBpediaNE>

```

```

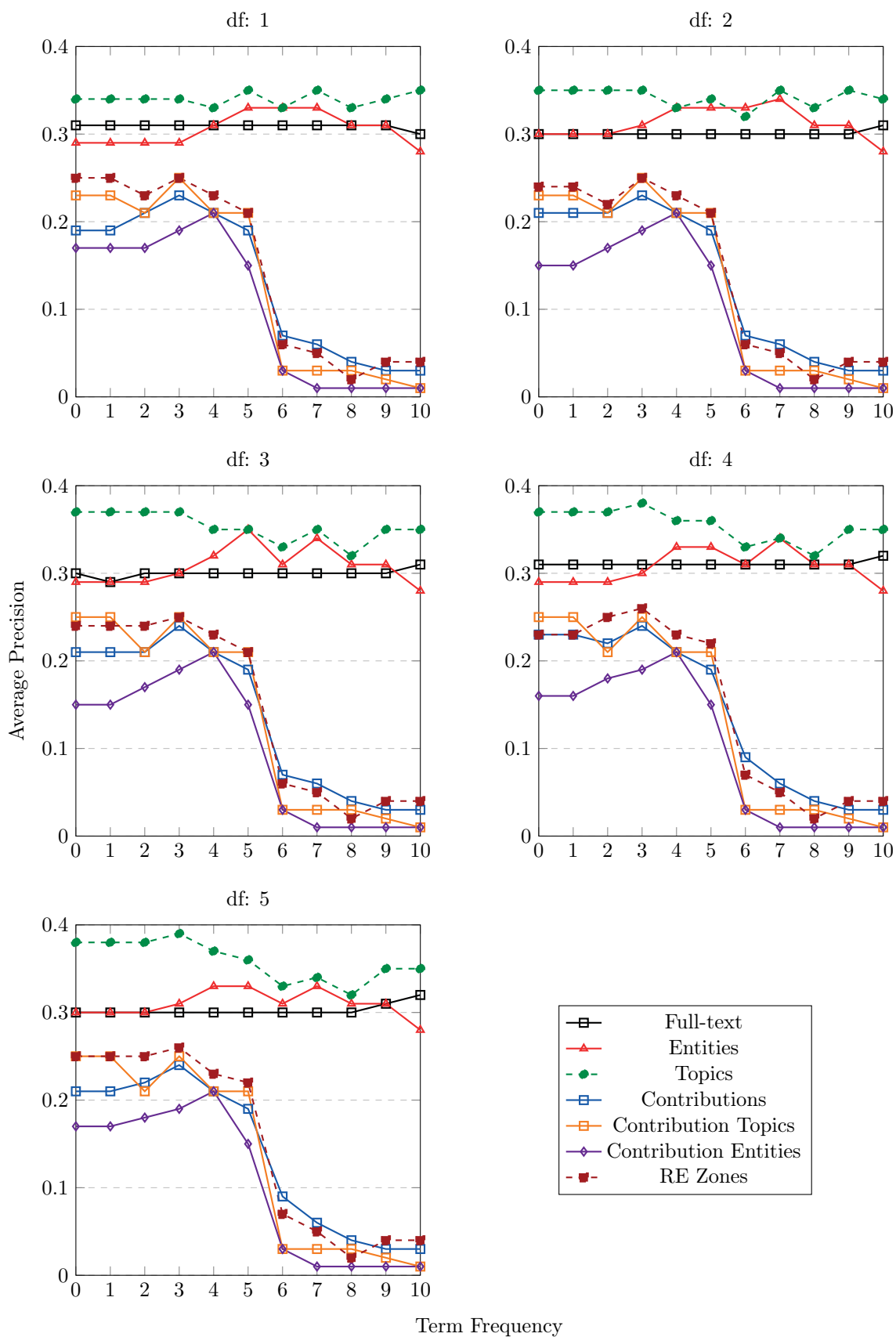
84  a <map:Mapping> ;
85  map:hasMapping <ex:GATEContentMapping>,
86  <ex:GATEStartOffsetMapping>,
87  <ex:GATEEndOffsetMapping>,
88  <ex:GATELODRefFeatureMapping> ;
89  map:baseURI <http://semanticsoftware.info/lodexporter/> ;
90  map:type <pubo:LinkedNamedEntity> ;
91  map:GATEtype "DBpediaNE" .
92
93 <ex:GATETitle>
94  a <map:Mapping> ;
95  map:hasMapping <ex:GATEContentMapping> ;
96  map:baseURI <http://semanticsoftware.info/lodexporter/> ;
97  map:type <doco:Title> ;
98  map:GATEtype "Title" .
99
100 <ex:GATEAuthor>
101  a <map:Mapping> ;
102  map:hasMapping <ex:GATEContentMapping>,
103  <ex:GATEStartOffsetMapping>,
104  <ex:GATEEndOffsetMapping>,
105  <ex:GATELODRefFeatureMapping>,
106  <ex:GATEFirstnameFeatureMapping>,
107  <ex:GATELastnameFeatureMapping> ;
108  map:baseURI <http://semanticsoftware.info/lodexporter/> ;
109  map:type foaf:Person ;
110  map:GATEtype "Author" .
111
112 <ex:GATEAffiliation>
113  a <map:Mapping> ;
114  map:hasMapping <ex:GATELocatedInFeatureMapping>,
115  <ex:GATELocationURIFeatureMapping>,
116  <ex:GATEContentMapping>,
117  <ex:GATEStartOffsetMapping>,
118  <ex:GATEEndOffsetMapping> ;
119  map:baseURI <http://semanticsoftware.info/lodexporter/> ;
120  map:type foaf:Organization ;
121  map:GATEtype "Affiliation_univ" .

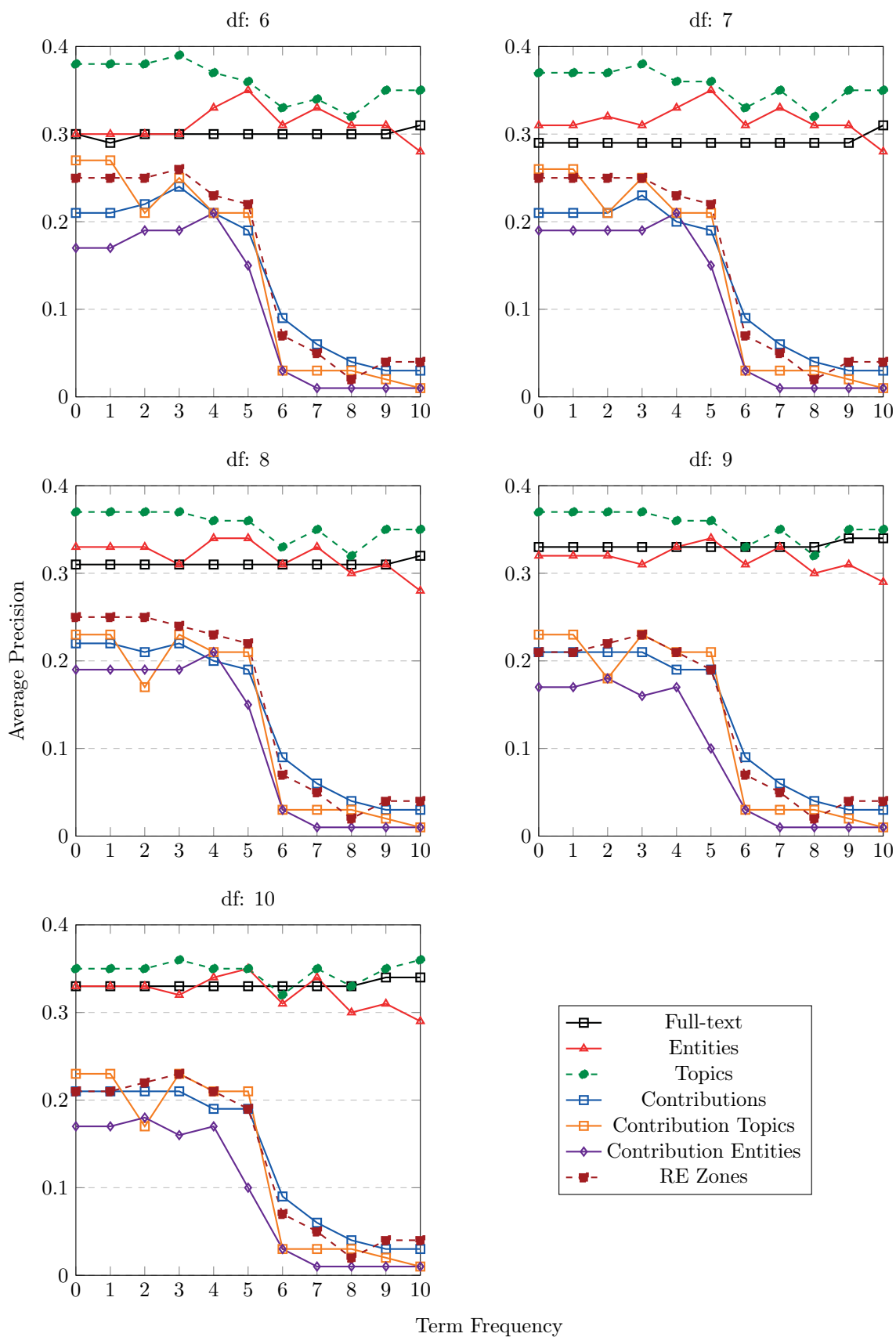
```

Appendix G

Semantic Vectors Evaluation Results

The diagrams shown in this section are the results of our experiments to find the best performing configuration for term-vector construction, as well as the optimized values for the minimum *tf* and *idf* metrics in Solr.





Appendix H

Solr Configuration Schema

The complete Solr version 1.6 core configuration settings for the semantic vector-based experiments reported in Section 8.4 are shown below:

Field name	Type	Multi-valued	Stored	Indexed	Term Vectors
id	string	✗	✓	✓	✗
fulltext	text_general	✗	✓	✓	✓
entity	lod	✓	✓	✓	✓
topic	text	✓	✓	✓	✓
claim	text	✓	✓	✓	✓
claim_uri	lod	✓	✓	✓	✓
contribution	text	✓	✓	✓	✓
contribution_uri	lod	✓	✓	✓	✓
*_txt	text_general	✓	✓	✓	✓

Field Type	Language Analyzers
string	inherit from solr.StrField class
text	solr.StandardTokenizerFactory, solr.StopFilterFactory, solr.LowerCaseFilterFactory, solr.SnowballPorterFilterFactory
text_general	solr.StandardTokenizerFactory, solr.StopFilterFactory, solr.LowerCaseFilterFactory, solr.SnowballPorterFilterFactory
*_txt	solr.StandardTokenizerFactory, solr.StopFilterFactory, solr.LowerCaseFilterFactory, solr.SnowballPorterFilterFactory
lod	solr.PatternReplaceCharFilterFactory (pattern="http:\\/(\\w+)\\/(\\w+)\\/", replacement=""), solr.WhitespaceTokenizerFactory, solr.LowerCaseFilterFactory