DEPARTMENT OF INFORMATICS
UNIVERSITY OF FRIBOURG (SWITZERLAND)

Entity-Centric Knowledge Discovery for Idiosyncratic Domains

THESIS

Presented to the Faculty of Science of the University of Fribourg (Switzerland)
in consideration for the award of the academic grade of
Doctor scientiarum informaticarum

by

ROMAN PROKOFYEV

from

RUSSIA

Thesis No: 1970
UniPrint
2016

Accepted by the Faculty of Science of the University of Fribourg (Switzerland) upon the recommendation of Prof. Dr. Abraham Bernstein and Prof. Dr. Roberto Navigli.

Prof. Dr. Ulrich Ultes-Nitsche, president of the jury

Fribourg, June 10, 2016

Thesis supervisor

Prof. Dr. Philippe Cudré-Mauroux

Dean

Prof. Dr. Fritz Müller

“Sapere aude”
[Dare to know]

— Quintus Horatius Flaccus

Acknowledgments

First of all, I would like to thank my supervisor, Professor Philippe Cudré-Mauroux, for his invaluable guidance and support during my time at the eXascale Infolab group. I really enjoyed the working process and the environment he established in the group, which allowed us to produce high quality work and publish it in the top venues. Moreover, not only he guided me and the other members or the group with his professional advice, but also educated us about the various peculiarities of the life in Switzerland.

Likewise, I would like to express my sincere appreciation to the whole eXascale Infolab group, its former and current members. This was one of the best places I have ever worked in my life, with friendly and relaxed atmosphere. This group gathered many outstanding people, with whom I had pleasure to work with. Specifically, I would like to extend my gratitude to my brave officemate, Alberto Tonon, with whom I shared office during my entire PhD and who became my truly friend; to Gianluca Demartini, who were a postdoctoral researcher at the time of my doctorate, and from whom I learned a lot on how to properly conduct scientific experiments and write papers; to my other co-authors from within and outside of the group, including Djellel Difallah, Michael Luggen, Ruslan Mavlyutov, Loïc Vouilloz, Alexey Boyarsky, Oleg Ruchayskiy, Martin Grund, Michele Catasta and Karl Aberer.

Finally, I would like to thank my family for tremendous support and believing in me in all times. In particular, I am deeply grateful to my mother, who continuously invested a lot of effort in me and my brother's education despite all the difficulties we had. I am also especially thankful to my brilliant wife, who was always aspiring me to find the appropriate decision in moments of doubt.

Fribourg, 18 August 2016 Roman Prokofyev

Abstract

Technical and scientific knowledge is produced at an ever-accelerating pace, leading to increasing issues when trying to automatically organize or process it, e.g., when searching for relevant prior work. Knowledge can today be produced both in unstructured (plain text) and structured (metadata or linked data) forms. However, unstructured content is still the most dominant form used to represent scientific knowledge. In order to facilitate the extraction and discovery of relevant content, new automated and scalable methods for processing, structuring and organizing scientific knowledge are called for. In this context, a number of applications are emerging, ranging from Named Entity Recognition (NER) and Entity Linking tools for scientific papers to specific platforms leveraging information extraction techniques to organize scientific knowledge. In this thesis, we tackle the tasks of Entity Recognition, Disambiguation and Linking in idiosyncratic domains with an emphasis on scientific literature. Furthermore, we study the related task of co-reference resolution with a specific focus on named entities.

We start by exploring Named Entity Recognition, a task that aims to identify the boundaries of named entities in textual contents. We propose a new method to generate candidate named entities based on n-gram collocation statistics and design several entity recognition features to further classify them. In addition, we show how the use of external knowledge bases (either domain-specific like DBLP or generic like DBPedia) can be leveraged to improve the effectiveness of NER for idiosyncratic domains. Subsequently, we move to Entity Disambiguation, which is typically performed after entity recognition in order to link an entity to a knowledge base. We propose novel semi-supervised methods for word disambiguation leveraging the structure of a community-based ontology of scientific concepts. Our approach exploits the graph structure that connects different terms and their definitions to automatically identify the correct sense that was originally picked by the authors of a scientific publication. We then turn to co-reference resolution, a task aiming at identifying entities that appear using various forms throughout the text. We propose an approach to type entities leveraging an inverted index built on top of a knowledge base, and to subsequently re-assign entities based on the semantic relatedness of the introduced types.

Finally, we describe an application which goal is to help researchers discover and manage scientific publications. We focus on the problem of selecting relevant tags to organize collections of research papers in that context. We experimentally demonstrate that the use of a community-authored ontology together with information about the position of the concepts in the documents allows to significantly increase the precision of tag selection over standard methods.

Abstract

Keywords: Knowledge Graphs, Knowledge Extraction, Named Entity Recognition, Entity Linking.

Zusammenfassung

Mit zunehmender Geschwindigkeit wird immer mehr technisch-wissenschaftliches Wissen angehäuft. Das Organisieren und Verarbeiten dieses wird immer problematischer, zum Beispiel beim Aufsuchen von relevanten bestehenden Arbeiten. Wissen wird heutzutage auf zwei Arten abgelegt, zum einen in unstrukturierter Form (Fliesstext) zum anderen in strukturierter Form (Metadaten, Linked Data). Der Anteil der unstrukturierten Inhalte bleibt aber der überwiegende Anteil wie wissenschaftliches Wissen repräsentieren wird. Um das Extrahieren und Auffinden von relevanten Inhalten, sowie um neue automatisierte und skalierbare Verarbeitungstechniken anzuwenden ist strukturiertes sowie organisiertes wissenschaftliches Wissen unabdingbar. Verschiedene Anwendungen die sich der Aufgaben annehmen sind in der Entstehung, diese Aufgaben bestehen aus Werkzeugen wie der Eigennamenerkennung (Named Entity Recognition oder NER) und der Entitäts Verlinkung für wissenschaftliche Arbeiten bis hin zu integrierten Plattformen welche mit Informationsextraktionstechniken wissenschaftliches Wissen organisieren. In dieser Arbeit behandeln wir die Aufgaben der Erkennung von Entitäten, der Disambiguation solcher und auch dem Verlinken von Entitäten innerhalb spezifischen Domänen, insbesonderen deren der wissenschaftlichen Literatur. Weiter behandeln wir verwandte Aufgaben der Koreferenz Auflösung mit dem Fokus auf Eigennamen.

Wir beginnen mit der Analyse der NER, welche das Ziel hat die Grenzen von Eigennamen in Fliesstext zu erkennen. Wir stellen eine neue Methode vor welche Kandidaten für Eigennamengrenzen basierend auf Statistiken von N-Grammkollokation findet, als auch entwickelten wir verschiedene Features um diese Entitäten weiter zu klassifizieren. Weiter zeigen wir wie man externe Wissensdatenbanken (domänenspezifische wie die DBLP oder allgemeiner Natur wie DBPedia) einsetzten kann um die Effektivität der NER für spezifische Domänen zu erhöhen. Anschliessend behandelten wir die Disambiguation der Entitäten welches normalerweise nach der NER angewandt wir um die Entitäten an eine Wissensdatenbank zu knüpfen. Wir stellen neue halb überwachte (semi-supervised) Methoden zur Disambiguation vor, welche die Struktur von kollaborativ geschaffenen Ontologien zu wissenschaftlichen Konzepte mit einbeziehen. Unser Ansatz verwendet die Graphenstruktur welche die verschiedenen Terme und deren Definitionen verbindet um automatisch die richtige Bedeutung zu identifizieren, welche durch den Autor der wissenschaftlichen Publikation gewählt wurde. Abschliessend behandeln wir die Koreferenz Auflösung bei welcher wir zusammenhängende Entitäten im selben Text mit unterschiedlicher Form identifizieren. Wir stellen einen Ansatz vor welcher Entitäten durch die Verwendung eines invertierten Indexes basierend auf einer allgemeinen Wissensdatenbank typisiert um anschliessend die Entitäten auf Grund der semantischen nähe

der gefunden Typen neu zuzuordnen.
Schlussendlich beschreiben wir eine Anwendung welche hilft Forschenden wissenschaftliche Publikationen aufzufinden und zu organisieren. Unser Fokus liegt auf der Auswahl von relevanten Schlüsselwörtern (tags) aus der Domäne definiert durch eine Sammlungen von wissenschaftlichen Dokumenten. Wir konnten experimentell aufzeigen, dass die Nutzung einer kollaborativ erstellten Ontologie verknüpft mit Informationen über die Position der Konzepte in den jeweiligen Dokumenten, die Präzision signifikant erhöht verglichen mit gängigen Methoden.

Stichwörter: Wissensgraphen, Wissensextraktion, Eigenamenerkennung, Named Entity Recognition, Entitäts Verlinkung.

Sommario

Una delle conseguenze del rapido sviluppo delle conoscenze tecniche e scientifiche è la produzione di quantità sempre maggiori di dati che, data la loro mole, diventano sempre più difficili da gestire ed elaborare, e.g., durante la ricerca di fonti bibliografiche. Nonostante questi dati possano essere strutturati (metadati o dati collegati) o non strutturati (testo libero), al giorno d'oggi gran parte della conoscenza scientifica è rappresentata con dati non strutturati. Questo motiva il pressante bisogno di nuovi metodi automatici e scalabili che semplifichino l'estrazione e la scoperta di nuova conoscenza scientifica. In questo contesto trovano spazio varie applicazioni tutt'ora oggetto di ricerca che includono, ad esempio, strumenti per il riconoscimento di entità nominate (Named Entity Recognition o NER), e per la connessione di entità presenti in documenti scientifici a diverse base di conoscenza (Entity Linking). Tali strumenti fanno abbondante uso di tecniche di estrazione di informazione per organizzare e gestire la conoscenza scientifica. In questa tesi trattiamo il riconoscimento (Entity Recognition), la disambiguazione (Entity Disambiguation), ed il collegamento (Entity Linking) di entità facenti parte di domini specifici, in particolare della letteratura scientifica.

Inizialmente affrontiamo il problema del riconoscimento di entità nominate, il cui scopo è identificare le frontiere delle entità nominate presenti in un dato contenuto testuale, proponendo un nuovo metodo per la generazione di nomi di entità a partire da statistiche sulle co-occorrenze di N-grammi. Mostriamo anche come basi di conoscenza esterne possono essere sfruttate per migliorare l'efficacia di sistemi di NER in domini specifici e come i metodi che descriviamo possono essere applicati sia a basi di conoscenza relative ad un dominio specifico (e.g., DBLP) che a basi di conoscenza generiche come (e.g., DBpedia). Successivamente sposteremo la nostra attenzione sulla disambiguazione di entità (Entity Disambiguation). Questo passaggio consiste nel collegare un frammento di testo rappresentante un'entità ad una voce presente in una base di conoscenza tenendo in considerazione che, a dipendenza del contesto, lo stesso frammento di testo può essere associato a diverse voci in essa contenute. In questo contesto proponiamo dei nuovi metodi semi-supervisionati per la disambiguazione di parole che sfruttano la struttura di una ontologia di concetti scientifici curata da una comunità di utenti. Il nostro approccio identifica il senso di un termine immerso in un dato contesto utilizzando un grafo che connette vari termini scientifici alle loro definizioni. Un altro tema chiave che approfondiamo è la risoluzione di coreferenze (Co-reference Resolution), che consiste nell'identificare entità che appaiono in un testo in diverse forme. In questo ambito proponiamo un approccio che sfrutta il tipo (o i tipi) di un'entità per selezionare le corrette forme testuali con cui essa viene rappresentata nel testo.

Per finire descriviamo un'applicazione atta a promuovere e gestire la ricerca di pubblicazioni scientifiche. In dettaglio, studiamo come etichettare articoli scientifici al fine facilitarne la catalogazione. I nostri esperimenti mostrano che l'utilizzo di un'ontologia curata da una comunità di utenti assieme ad informazione sulla posizione di vari concetti scientifici all'interno di articoli accademici aumenta notevolmente la precisione dell'algoritmo di etichettatura rispetto a metodi standard.

Parole chiave: Grafi Concettuali, Estrazione di Conoscenza, Riconoscimento di Entità Nominate, Collegamento di Entità.

Contents

List of Figures

List of Tables

1 Introduction

The way knowledge is represented in modern information systems is undergoing rapid changes that lead to significant shifts in the way core web applications operate. Over the last years, one important form of knowledge representation has been consolidated under the concept of so-called *Knowledge Graphs*. Knowledge Graphs are essentially entity-centric knowledge bases that store interlinked factual knowledge about entities. They embody entities along with their relationships and associated properties in a much richer form than traditional relational databases with rigid schemas. Relational database requires data to be stored in tabular form with a predefined schema, such as customer data, with names, ages and addresses. Entity-centric information, on the other hand, is schemaless and is best described by object-oriented data model with classes, subclasses, and instances, where instances can have pointers to other instances (ontologies).

Much of the valuable information in Knowledge Graphs comes from extracting it from unstructured and semi-structured (i.e., web pages) textual resources on a large scale. Nevertheless, they do not replace existing unstructured data sources. Instead, they act as a semantic layer on top of the them and continuously extract and store valuable information [33]. As compared to traditional full-text indexes, information extracted and stored in Knowledge Graphs spans beyond mere keywords. Entities can appear in various forms in a text and, more importantly, the same words may or may not refer to the same entity in different contexts, which makes searching entities using keyword indexes error-prone.

One of the most prominent examples of knowledge graphs in Computer Science community is DBpedia [17], which structures the data extracted from Wikipedia. Entities in DBpedia correspond to pages in Wikipedia, but contain structured semantic properties, such as dates and places of birth, profession, etc. for entities that represent people. The advent of DBpedia and other knowledge bases gave birth to a variety of novel and effective methods for tackling knowledge extraction and discovery tasks [149, 37]. Question answering systems have flourished and now make heavy use of factual properties about entities [154, 144, 124, 83], search result diversification can take types of entities into account to make results more diverse [153, 129], and, of course, various analyses on how entities are related to each other via

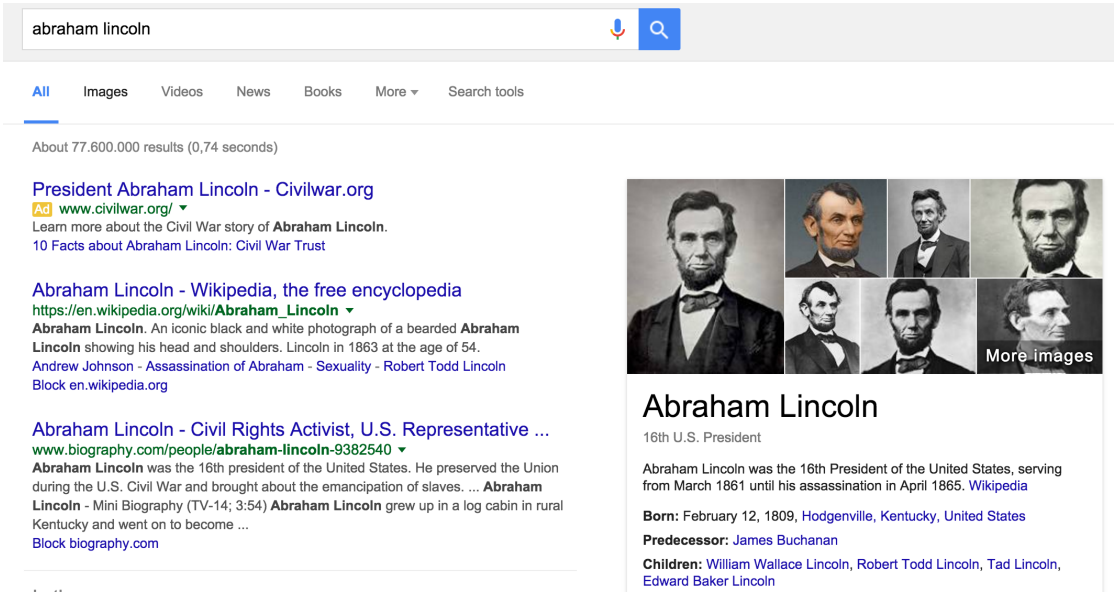

Figure 1.1 – The web interface of Google search with results from their Knowledge Graph (on the right).

property graphs [106, 60].

Many large companies also embarked on their own knowledge graph projects, namely the Knowledge Graph[1] from Google, Satori Knowledge Base[2] from Microsoft Bing, Facebook Open Graph[3] and IBM Concept Insights[4] that is currently used to extract concepts from articles in the ACM Digital Library. Such graph-based data representations currently back some of their main products. Google search, for instance now considers a query like “abraham lincoln” as more than just two distinct keywords but rather as a whole entity with properties and links to other entities (Figure 1.1). This provides significant business value for companies, since users can often find the necessary information directly on a search results page, thus potentially staying longer on the website.

While giant tech companies have the resources to build general purpose knowledge graphs to support their applications, such graphs are not sufficient for many idiosyncratic domains as they do not capture specific vertical knowledge. In the context of this thesis, we define idiosyncratic domains as vertical fields of knowledge having their own terminology. Examples of idiosyncratic domains include physics, chemistry, biomedicine, computer science, etc. Contrary to general knowledge domains, emerging sub-areas of idiosyncratic domains often lack well-established terminology in the knowledge graph, which makes them harder to analyze. Existing academic applications mainly focus on reference managers for organizing

[1] https://www.google.com/intl/bn/insidesearch/features/search/knowledge.html
[2] https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing
[3] https://developers.facebook.com/docs/opengraph
[4] http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/concept-insights.html

bibliography, such as Mendeley[5] and Zotero[6]. However, Knowledge graphs are likely to be of even greater importance for idiosyncratic domains, because they are harder to index, search and understand than general knowledge.

Therefore, it is worth investigating what methods and solutions are necessary to mine structure and knowledge from and for idiosyncratic domains. These methods range from identification of relevant entities in textual data, to disambiguating them among existing knowledge graph entities, to resolving multiple appearances of an entity inside a given text (see Chapter 2). Finally, it is equally important to search for innovative use cases of idiosyncratic knowledge that can, for instance, foster scientific research. This thesis improves on both state-of-the-art methods for solving existing tasks related to the construction of knowledge graphs, as well as elaborates on new innovative use cases for idiosyncratic data. These topics are introduced in more detail below.

1.1 Knowledge extraction

Knowledge extraction is a complex task that involves identifying entities and relationships from unstructured or semi-structured documents (e.g. PDFs, Rich Text Documents, HTML web pages, etc.), as well as structured sources (databases). The procedure typically involves multiple steps, depending on the required level of granularity of the extraction process and on the application. Individual steps include tasks such as Named Entity Recognition (Section 1.1.1), Entity Disambiguation (Section 1.1.2), extraction of relations between entities, coreference resolution (Section 1.1.3), fine-grained typing of entities, etc. Furthermore, these steps can be interconnected. For example, relation extraction and coreference resolution steps depend on the entity recognition and disambiguation; also, high-precision coreference resolution can contribute back to improving the quality of entity recognition. The state-of-the-art solutions to most of these problems require labeled data to learn from which can be hard to obtain depending on the application (see Section 2.3).

1.1.1 Named Entity Recognition

Named Entity Recognition (NER) is the task of recognizing an entity mention in a text, and possibly assigning a type to this entity, such as “Person” or “Location”. Most NER systems also treat various numerical expressions as entities, for example dates and times, temperature and volume values, etc. Thus, it is considered to consist of two separate parts: identification of an entity boundary (entity mention) and assignment of a type to an entity. The entity to recognize, however, does not have to be present in any knowledge base in order to be identified.

Despite the large amounts of publications on the topic over the last decade, NER is far from being solved. The most important issues include reducing the amount of manual labor

[5] https://www.mendeley.com
[6] https://www.zotero.org/

required to train supervised classifiers, multi-domain robustness of the methods (or lack of effective NER systems for certain domains, such as idiosyncratic ones), and fine-grained type assignment (e.g., "Politician" is a fine-grained type as compared to "Person").

Nested entities and abstractions represent another set of challenges in NER. For example, in a phrase "brands such as BMW, Volkswagen, and Audi", there are three entities that represent car manufacturers, but also the whole phrase can refer to an entity "Automotive Industry". Likewise, in a phrase "automotive company created by Henry Ford in 1903", there is an entity "Henry Ford" and a date "1903", but the whole phrase also refers to an entity "Ford Motor Company".

The major issues with NER on idiosyncratic data are the large variety of such named entities and emerging entities, which means they are not yet present in many knowledge bases. This leads to an impossibility to use dictionaries of entity labels, as for the case of well-known geographical locations or person names. Another problem is that idiosyncratic entities are often not syntactically distinguishable in text, using title-casing for example, which means that a NER system basically needs to somehow identify candidate entities directly from text and then classify whether it is an entity or not. This work tackles the problem of NER in idiosyncratic domains in Chapter 3, on an example of Physics and Computer Science domains.

1.1.2 Named Entity Disambiguation

The purpose of the Entity Disambiguation task is to associate (link) entities from an existing knowledge base to an entity mention. Thus, identifying entity mentions in the text is a necessary prerequisite step for this task. The joint process of finding entity mentions and the subsequent disambiguation step is referred to as Entity Linking.

The problem of disambiguation arises when labels of multiple entities from a knowledge base match a given entity mention. Candidate entities for a given mention are typically generated from the labels of such entities that are either already available in a knowledge base, or extracted from a manually annotated textual corpus, such as Wikipedia. Consider the following sentence: "Roosevelt has consistently been ranked by scholars as one of the greatest U.S. presidents". Here, the entity mention "Roosevelt" is ambiguous, since it can refer to a multitude of entities (people, places, etc.); even if a system is somehow able to understand that the mention refers to a person who is a president of the United States, there are still two presidents whose last names are Roosevelt.

Major classes of ambiguous mentions include first and last names of people when they appear separately, as well as names of geographical locations, e.g., cities. However, in general it is also incorrect to blindly link a mention to a single candidate entity, since this mention can also be a so-called NIL entity (not an entity), which can often happen with titles of artistic works, such as titles of movies, songs and books.

Another large class of ambiguous mentions are acronyms that ubiquitously present in id-

iosyncratic content. This work tackles the problem of acronym disambiguation for the case of Computer Science and Biomedical domains in Chapter 4.

1.1.3 Coreference resolution

Coreference resolution aims to identify additional mentions of entities in a text that are made via coreferences. Coreferences can be pronouns for entities of type "Person", but can also be other nouns or phrases, such as phrases with words "this" or "the" as a qualifier, e.g. "I have visited the Rushmore Mountain. This mountain is very beautiful."

Solving coreference resolution allows to collect additional contexts in which entities are used, thus helping to better understand unknown and emerging entities, their types, their properties or collect personal opinions about them [9, 43, 6]. This thesis tackles the problem of coreference resolution with a particular focus on noun-based coreferences using ontology-based methods in Chapter 5.

1.2 Knowledge Extraction and User-Generated Content

Increase of global access to the Internet led to the accelerated growth of user-generated content, including textual content. English Wikipedia, for example, has currently grown to more than 10 million articles. Most of the articles contain one or more links to other Wikipedia entities (pages), all added by numerous volunteers around the world. Currently, the total number of such links in Wikipedia exceeds 78 millions. This data allows to extract textual labels for named entities at scale, significantly improving recall of modern Entity Recognition and Entity Linking systems [92, 89]. Furthermore, this data can be used to calculate usage statistics for labels and entities, which allows to compute both conditional probabilities of entities for a given label, and probabilities of labels for a given entity. Subsequently, since natural text exhibits more or less the same properties as Wikipedia articles, these probabilities represent one of the most powerful baselines for Entity Linking systems [51].

The amount of information has also grown in idiosyncratic domains, such as the increased number of scientific publications on arxiv.org[7]. The content, however, typically does not contain well-marked entities and can be considered as semi-structured texts. Nevertheless, one can develop statistical models to understand these vast amounts of unstructured data; for example, one can build n-gram models for extracting key phrases from a single document [73, 142], or it is possible to identify and extract entities from collections of similar documents. This thesis elaborates on extracting named entities from idiosyncratic data in Chapter 3.

1.3 The ScienceWISE System

As we mentioned earlier, scholar applications mostly focus on organization of bibliography by means of manual categorization. Such systems, however, do not perform deep semantic analysis of the vast amounts of available scientific literature. Over the course of this thesis, we have participated in research and development of a conceptually novel system called ScienceWISE[8] [3] (Figure 1.2). The goal of the system is to facilitate scientific research by providing easy ways to navigate through and discover scientific literature, as well as organize collections of related work on any topic. Newly arrived articles are ranked according to the interest of users, which is based on concepts. The core of the system is an expert-curated ontology of scientific concepts, which is essentially a knowledge graph that holds scientific concepts (entities), their labels, and various relationships among them. This ontology constantly evolves and grows with the help of the researchers that use the system daily. When researchers add articles to their collections, the system suggests a list of concepts to tag the paper with, including candidate concepts that are not part of the knowledge graph, but that are automatically added when people choose them. We detail the workflow of the system and investigate the problem of recommending concepts for better tagging of scientific articles in Chapter 6.

1.4 Summary of Contributions

The goal of this thesis is to *develop better the methods and applications to mine structured data and knowledge from and for idiosyncratic domains.* In practice, we implement several systems that can act independently or be parts of a larger knowledge extraction pipeline.

[7] http://arxiv.org/stats/monthly_submissions
[8] http://sciencewise.info

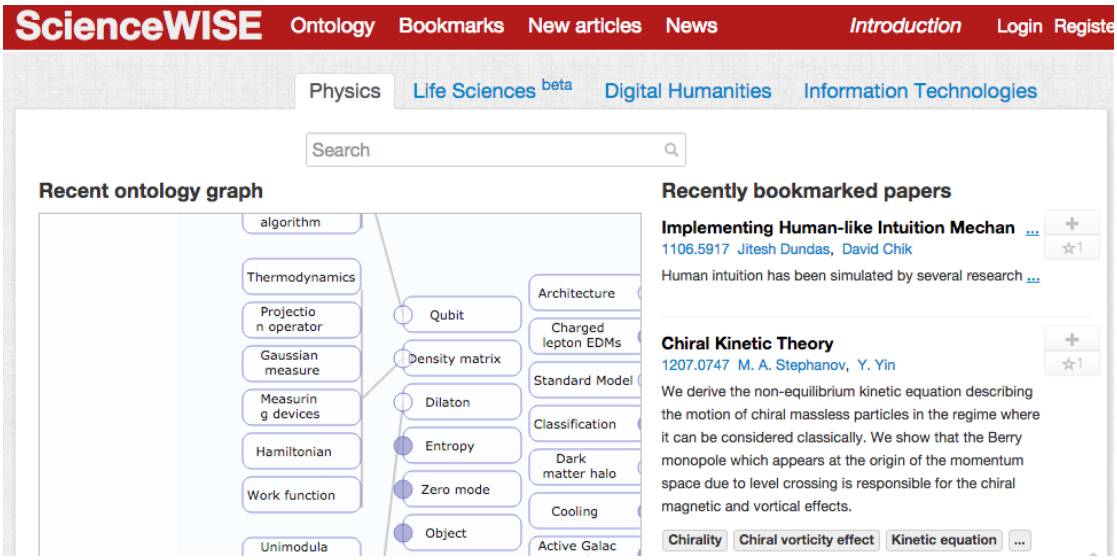

Figure 1.2 – Interface of the ScienceWISE System.

In the following, we detail our contributions and list the associated conference and journal papers we have published along our research work. We tackle the previously discussed tasks of NER, entity disambiguation in the context of idiosyncratic knowledge, as well as few other methods related to general knowledge graph construction.

A) *Knowledge extraction* is a multi-step process where the number and the steps themselves can vary depending on the use case. In that context, our research focused on two primary stages of almost any knowledge extraction pipeline, namely Named Entity Recognition and Entity Disambiguation, and on the Coreference Resolution task, which allows to gather even more information about entities:

Named Entity Recognition: In this first work, we proposed novel approaches for NER on idiosyncratic document collections (such as scientific articles) based on inspection and classification of n-grams. We designed and evaluated several entity recognition features—ranging from well-known part-of-speech tags to n-gram co-location statistics and decision trees—to classify candidates. We evaluated our system on two test collections created from a set of Computer Science and Physics papers and compared it against state-of-the-art supervised methods. Experimental results showed that a careful combination of the features we propose yield up to 85% NER accuracy over scientific collections and substantially outperforms state-of-the-art approaches such as those based on maximum entropy.

Roman Prokofyev, Gianluca Demartini and Philippe Cudré-Mauroux. "Effective named entity recognition for idiosyncratic web collections". Proceedings of the 23rd international conference on World Wide Web. ACM, 2014.

Entity Disambiguation: In another work, we proposed novel semi-supervised methods to entity disambiguation leveraging the structure of a community-based ontology of scientific concepts. Our approach exploits the graph structure that connects different concepts and their definitions to automatically identify the correct sense that was meant by the authors of a scientific publication. Experimental evidence over two different test collections from the physics and biomedical domains shows that the proposed method is effective and outperforms state-of-the-art approaches based on feature vectors constructed out of term co-occurrences as well as standard supervised approaches.

Roman Prokofyev, Gianluca Demartini, Alexey Boyarsky, Oleg Ruchayskiy and Philippe Cudré-Mauroux. "Ontology-based word sense disambiguation for scientific literature". Proceedings of the 35th European conference on Advances in Information Retrieval. Springer-Verlag, 2013.

Coreference resolution: In our latest work, we tackled the problem of resolving coreferences in textual content by leveraging Semantic Web techniques. Specifically, we focused on noun phrases that reference identifiable entities. We first applied state-of-the-art techniques to extract entities, noun phrases, and candidate coreferences. Then, we proposed an approach to type noun phrases using an inverted index built on top of a Knowledge Graph (e.g., DBpedia). Finally, we used the semantic

relatedness of the introduced types to improve the state-of-the-art techniques by splitting and merging coreference clusters. We evaluate our system on CoNLL datasets, and show how our techniques consistently improve the state of the art in coreference resolution.
Roman Prokofyev, Alberto Tonon, Michael Luggen, Loic Vouilloz, Djellel Eddine Difallah and Philippe Cudré-Mauroux. "SANAPHOR: Ontology-Based Coreference Resolution". Proceedings of the 14th International Semantic Web Conference. Springer International Publishing, 2015.

B) The last part of the thesis presents a novel application for idiosyncratic data that helps researchers to organize and discover new articles, ScienceWISE. We focus specifically on:

Tag recommendation: where we tackled the problem of improving the relevance of automatically selected tags in large-scale ontology-based information systems. Contrary to traditional settings where tags can be chosen arbitrarily, we focused on the problem of recommending tags (e.g., concepts) directly from a user-driven ontology. We compared the effectiveness of a series of approaches to select the best tags ranging from traditional IR techniques such as TF/IDF weighting to novel techniques based on ontological distances and latent Dirichlet allocation. All our experiments are run against a real corpus of tags and documents extracted from the ScienceWise portal, which is connected to arXiv and is currently used by growing number of researchers.
Roman Prokofyev, Alexey Boyarsky, Oleg Ruchayskiy, Karl Aberer, Gianluca Demartini and Philippe Cudré-Mauroux. "Tag recommendation for large-scale ontology-based information systems". Proceedings of the 11th International Semantic Web Conference. Springer International Publishing, 2012.

1.4.1 Additional Contributions

In addition to the core contributions of this thesis, which are listed above, we also published the following pieces of work related to knowledge extraction and discovery.

1. We contributed to *TRank*[138], a system that ranks types of a known entity given the text this entity appears in. In that work, we extended the methods used by the system with new methods to find the most relevant entity type based on collection statistics and on the knowledge graph structure interconnecting entities and types.
Alberto Tonon, Michele Catasta, Roman Prokofyev, Gianluca Demartini, Karl Aberer and Philippe Cudré-Mauroux. "Contextualized ranking of entity types based on knowledge graphs". Journal of Web Semantics: Special Issue on Knowledge Graphs. Elsevier, 2016.
2. In another work, we tackled the task of detecting and correcting grammatical errors of non-native English speakers. We proposed a series of approaches for correcting prepositions that leveraged n-gram statistics, association measures, and machine learning techniques. We evaluated the effectiveness of our approach on two test collections

created from a set of English language exams and StackExchange forums.
Roman Prokofyev, Ruslan Mavlyutov, Martin Grund, Gianluca Demartini and Philippe Cudré-Mauroux. "Correct Me If I'm Wrong: Fixing Grammatical Errors by Preposition Ranking". Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM, 2014

1.5 Outline

The rest of this thesis is organized as follows. Chapter 2 reviews relevant work and existing methods tackling typical knowledge extraction and discovery tasks. Next, we delve into Named Entity Recognition (NER) for idiosyncratic domains in Chapter 3. We combine n-gram statistics from document collections and the structure of connections in domain-specific ontologies to develop a robust NER system. In Chapter 4 we present and evaluate an ontology-based entity disambiguation method for the case of scientific documents, where we exploit connections between entities in an ontology graph to disambiguate entity mentions. Chapter 5 continues our exploration of knowledge extraction tasks with the task of Coreference Resolution. We design a method that regroups entity mentions in coreference clusters based on their semantic properties on top of a state-of-the-art coreference resolution system. Finally, Chapter 6 gives a brief overview of the ScienceWISE system that employs knowledge extraction techniques to help researchers organize and discover scientific literature. In particular, we focus on recommending entities for tagging scientific articles. We conclude in Chapter 7 which summarizes our main findings and outlines future directions, as well as give an outlook for future developments on knowledge graphs.

2 Background in Knowledge Extraction and Discovery

2.1 Introduction

Knowledge extraction from unstructured data sources dates back to 1970s, the early days of Natural Language Processing (NLP), where knowledge extraction was motivated by the need to provide real-time news for financial traders [65, 7] and by the U.S. Defense Advanced Research Projects Agency (DARPA) which wanted to automate tasks such as searching news for possible links to terrorism [26]. Since then, knowledge extraction has evolved from basic tasks such as segmentation and part-of-speech tagging to more sophisticated ones that usually involve multiple steps, such as Named Entity Recognition, Entity Disambiguation, Co-reference Resolution, Relation Extraction, etc. [47].

In the following, we give some background on algorithms and systems leveraging knowledge extraction, and the techniques used for Named Entity Recognition, Entity Disambiguation and Coreference Resolution, which are the main themes covered in this thesis. Additional related work is also covered in the corresponding chapters.

2.2 Part-of-Speech Taggers

Part-of-Speech (POS) tagging is the task of annotating words in text with corresponding POS tags (nouns, verbs, etc.). The challenge in POS tagging lies in the fact that the same words can correspond to different parts of speech depending on the context (e.g., "to tag on facebook" – where "tag" is a verb vs. "part of speech tag" – where "tag" is a noun). POS tagging is important for many knowledge extraction tasks as POS tags are often used as features for more sophisticated problems. For example, they are used as a basis for dependency parsing [23]; NER systems also use POS tag patterns to mine new entity candidates [107, 112], while coreference resolution systems use POS tags to find co-referring entities [115, 114].

POS tagging is one of the earliest problems in information extraction. Early systems for POS tagging were developed in 1960s when the first large (~1M words) manually annotated corpus

was created at Brown University. Currently, POS tagging for English is considered to be solved as modern state-of-the art systems achieve ~97% accuracy. The top-performing methods for POS tagging are based on Maximum Entropy Models [141] and Neural Networks [25].

2.3 Named Entity Recognition

As we discussed in Chapter 1, Named Entity Recognition (NER) is the task of recognizing entity mentions in a text, and subsequently assigning types to these entities. Here, we refer to an entity as an element of text that exists by itself, "concretely or abstractly, physically or not"[1], irrespectively of its presence in any knowledge base. An entity can be a person, an organization, or a location, although particular types of entities can vary (see Section 2.3.4). NER provides immense value to a multitude of applications, including search engines, relation extraction and Q&A systems. Search engines use NER to identify and index entities in documents and match them with entities identified in search queries to provide more accurate search results [50, 5]. Relation and fact extraction methods typically establish relationships between entities rather than arbitrary words [10], e.g., for the "place of birth" relation, the subject is typically an entity of type "person", while the object is an entity of type "location". Finally, NER had a positive impact on Q&A systems, improving the quality of answers [95, 140], as answers to many questions are often named entities.

2.3.1 Supervised NER

Supervised learning currently remains the dominant technique for NER [97]. Supervised NER models are trained on manually labeled corpora, like the CoNLL corpus [137], where texts are marked with entity mentions and their associated types. Such methods are based on features that are automatically extracted from labeled corpora, such as words surrounding entities, POS tags patterns, character n-grams and word capitalization patterns [74, 40, 86]. The most prominent learning techniques include Maximum Entropy Classifiers [19], Conditional Random Fields [40] and Neural Networks [25].

2.3.2 Semi-Supervised NER

As we discussed in the previous section, supervised NER requires large amounts of manually labeled data, which can be hard and expensive to obtain. This poses a problem for NER in specific domains of knowledge, such as science, technology and their sub-domains, since there are no large annotated corpora to train NER systems on. To address this issue, some researchers have focused on Semi-Supervised NER, where a method first receives a small amount of so-called seed entities, finds discriminative context cues for them, and then train a classifier to identify new entities with similar cues [97, 150, 72]. These contextual cues can be words, phrases or templates (e.g. "episodes of [TV program]") appearing together with an

[1] https://en.wikipedia.org/wiki/Entity

entity of a particular type [150]. Wikipedia is typically used as a source of training data.

2.3.3 Unsupervised NER

Another direction of research is completely unsupervised NER, which also aims to solve the problem of getting obtaining labeled data. Many approaches in this context stem from the fact that the same entities tend to appear in documents on similar topics more often than other common nouns. Examples include studies on news articles [131], targeted twitter streams [80] and on scientific papers on the same topic [112]. Other approaches focus on identifying entity types based on co-occurrence of words in the context of an entity, where co-occurrence can be any statistical metric, such as Pointwise Mutual Information [36]. For instance, "London" probably co-occurs with the word "city" more often than with the word "country". This is similar to the approaches in Section 2.3.2, with the difference that the initial list of entities is also extracted automatically, using domain-independent extraction patterns [36].

2.3.4 Domain-specific NER

Robust performance of NER systems across different domains of knowledge presents one of the major challenges of Entity Recognition [118]. Supervised NER requires labeled data for each new domain. In addition, the domain itself can be loosely defined. While it is possible to reduce the amount of labeled data needed with domain specific knowledge such as lists of entity labels and their types, they are not always available. Besides the high costs associated with the manual annotation of the training data, supervised NER also raises the problem of domain-dependent classifiers; For instance, models trained for news articles might not perform well on other types of documents [108]. Furthermore, in many technical domains entities are often not capitalized in texts, which significantly lowers the recall of NER systems trained with capitalization pattern features [112]. In this context, we believe that multi-step systems are the most effective ones. One part of the system takes advantage of entities that are available in knowledge bases and identifies them in text, effectively performing Entity Linking (see Section 2.4), while another part discovers new entities based on properties of the language, such as entity templates or phrase co-occurrence statistics. This thesis presents such a NER system in Chapter 3.

2.4 Entity Linking and Disambiguation

Entity Linking (EL) corresponds to the task of identifying entity mentions in a text and associating them with their corresponding entities from a knowledge base. EL is similar to NER in the way that entities need to be identified in a text, but differs from it in that the disambiguation of entities has to be performed. For instance, NER systems might identify an entity "Roosevelt" and assign a type "Person" to it, but EL systems have to explicitly say which "Roosevelt" the text is referring to.

Thus, EL consists of the two following steps:

- Identification of entity mentions in a text;
- Linking entity mentions to corresponding entities in a knowledge base (Entity Disambiguation).

The identification step can be performed either through a NER system or based on string matching approaches using the labels of entities that are available in the knowledge base. A combination of these methods is also possible, since NER systems are usually not able to identify entity mentions of all types. A special case of the disambiguation step is linking an entity mention to a so-called NIL entity. NIL refers to the case where a mention of an entity is found correctly, but this entity is not present in a target knowledge base. In this case, EL is similar to NER, except it does not need to assign a type to the NIL entity.

EL plays a key role in a wide range of applications, such as search engines and Q&A systems, in a similar way to NER. With EL, however, an entity mention is linked to a knowledge base/graph entry, thus additional properties of this entity can be leveraged. For instance, search engines could benefit by computing most popular properties and related entities to present on search result pages. Another possibility is to compute timelines of entities, for example to display important events in actors' careers [6].

2.4.1 Local models

Entity Linking attracted attention with the creation of Wikipedia, which became the world's largest encyclopedia in 2007 with over two million articles[2]. Naturally, articles in Wikipedia became associated with entities in a broad sense. Early works on the topic focused on so-called Local models that make use of textual context surrounding entity mentions to disambiguate them. "Wikify!" [89] and "Learning to Link with Wikipedia" [92] are considered to be the earliest works on EL in that area. "Wikify!" focuses on identifying and linking only the most important (for a given text) named entities, similar to how humans annotate Wikipedia articles. It uses TF-IDF and Key-phraseness metrics to compute the importance of candidate entity mentions in a document and then rank candidate mentions according to it. On the other hand, in [92], the authors propose a method to identify all possible entities in a document using a machine learning classifier that takes into account both the probability of a mention being an entity and the mention context to make a decision.

For Entity Disambiguation, both [89] and [92] use a machine learning classifier that considers features for each entity mention and entity candidate, and predicts whether this candidate is correct. In [89], features are based on similarity statistics between the mention context and the Wikipedia page of an entity, including words surrounding an entity mention and their POS

[2] https://en.wikipedia.org/wiki/Wikipedia

tags, and entity-specific words extracted from Wikipedia pages using co-occurrence statistics of an entity and a given word. Features in [92] are based: 1) on the prior probability of a phrase being a certain entity and 2) on the relatedness metric that measures the similarity between a target document and an entity document (Wikipedia page) based on the sets of unambiguous entities inside these two documents.

2.4.2 Global models

Global models expand the notion of a context to the assumption that all entities within a document are coherent around the subject of the document. Such models try to jointly disambiguate entities in a document by finding assignments of entities to entity mentions with maximum semantic coherence. Cucerzan [27] uses a vector space model in which entities (Wikipedia pages) and documents are represented as vectors of entities. The disambiguation process then maximizes the similarity between the document vector and the entity vectors in terms of scalar products. Similarly, DBpedia Spotlight [88] uses weighted cosine similarity to disambiguate entities. The GLOW system [120] further improves global approaches by introducing a machine learning classifier that predicts whether an entity mention has to be linked to the NIL entity, thus pruning the list of entity mentions. Kulkarni [77] combines both local and global scores and models the EL process as an optimization problem that maximizes the sum of these scores. In general, global approaches achieve better disambiguation accuracy than local ones.

2.4.3 Graph-based methods

Finally, graph-based approaches [61, 56, 96, 51] establish connections between entities and entity mentions in a graph and use graph-based metrics to find the best possible disambiguations. They are similar to the global models in that they also assume entities in a document should be related to each other. The disambiguation approach in [61] constructs a weighted graph with connections between entities and entity mentions, and computes a dense subgraph such that it contains exactly one mention-entity edge for each mention. “Babelfy” [96] uses random walks to compute the semantic signatures of entities for the whole knowledge base. The disambiguation is then performed by choosing an entity with the maximum normalized weighted degree of the densest subgraph. REL-RW [51] adopts a similar approach based on random walks, but computes semantic signatures of entities and documents on a smaller graph on the fly. Then, it selects the entity with the maximum similarity score for a given mention.

Most of the graph-based approaches integrate both local (prior probability, context similarity) and global (coherence among entities in a document) measures to the weights in the graph. Graph-based methods currently provide the highest disambiguation accuracy for EL task on various datasets.

2.5 Coreference resolution

Coreference resolution refers to the task of identifying all appearances of an entity in a given text, not only via direct mentions, but also via coreferences. Typical coreferences used in text are pronouns (he, she, it, etc.), nouns with qualifiers like “this” or “the” (the president, this country), or parts of a full entity name. This is different from the EL task, which is limited by the entity labels that are present in a knowledge base.

Coreference resolution is closely related to the NER and EL tasks and allows to identify more entity mentions inside a given text. This can help to gather additional factual information about an entity, such as more precise types and other relations found in a text [43, 6].

Coreference resolution has a long history in research [103]. Current state-of-the-art coreference resolution system consist of multiple steps (“sieves”) that merge co-referring mentions according to deterministic rules [114]. Mentions are automatically generated by selecting all noun phrases, pronouns, and named entities. Each sieve then decides whether to merge a group of mentions to its best antecedent. The multi-step composition of such systems allows to easily extend them by incorporating additional sieves.

One of the typical approaches to coreference resolution is to incorporate additional semantic knowledge on top of a syntactic coreference resolution system. Works in this direction include augmenting machine learning based coreference resolution systems with features extracted from Wikipedia and WordNet [109], using types and labels from the YAGO ontology to improve matching between co-referring mentions [116], and using word co-occurrence statistics from a Web n-gram corpus to correctly match mentions and their antecedents [13]. Haghighi *et al* [52] describe a three-step method consisting of a syntactic parser that extracts mentions and potential antecedents, a semantic module that measures the compatibility of headwords and individual names, and a final step which assigns mentions to antecedents. In a follow-up work, the authors introduce a generative model assuming that every mention of an entity has a latent type that generated it [53].

Coreference resolution also received increased attention in idiosyncratic domains, including medical and clinical data for discourse-level analysis of clinical documents [71, 70, 29].

Many recent works experimented with integrating EL to increase the performance of coreference resolution systems. One approach is to enhance linked entities with additional attributes from knowledge bases and use them to learn the semantic compatibility of co-referring mentions. Ratinov and Roth [119] use GLOW [120] to link entities and attributes extracted from Wikipedia page categories. They utilize machine learning classifier on top of traditional multi-sieve systems to predict the pairwise compatibility of co-referring mentions based on extracted attributes. Similarly, NECo [54] extracts frequent attributes from Freebase and Wikipedia to add extra sieves that merge groups of co-referring mentions based on attribute matching. In this thesis, we pursue a similar approach for improving coreference resolution using both splitting and merging of groups of co-referring mentions on top of a state-of-the-art system

in Chapter 5.

2.6 Conclusions

Knowledge extraction has been intensively studied over the past years.One of the major trends in that context are Knowledge Graphs, entity-centric data structures that aggregate information from a variety of both structured and unstructured sources. In the light of recent advances in knowledge extraction techniques, it become possible to fuse existing knowledge bases with the knowledge extracted directly from unstructured data sources, forming large-scale probabilistic knowledge graphs [33]. Such large graphs can provide immense support for various downstream applications, including question answering or entity-based search, and even serve as knowledge engines for robots [128].

The contributions of this thesis are mainly technical; We design new algorithms to extract knowledge more effectively from new domains, and experimentally evaluate the results of our Entity Recognition, Entity Disambiguation and Co-reference resolutions methods in various setups. We believe that the results obtained in this context will be beneficial in developing better knowledge extraction systems in the future.

In the following chapter, we start by exploring the task of Named Entity Recognition, one of the main building blocks of any Knowledge Extraction System, with a particular focus on an under-explored domain of idiosyncratic data.

3 Named Entity Recognition for idiosyncratic collections

3.1 Introduction

While recent approaches to online Named Entity Recognition (NER) have become quite efficient and effective, they still do not perform equally well on all domains, leaving out some application scenarios from entity-centric information access. For highly-specialized domains such as academic literature, online information systems performing search, bookmarking, or recommendations are still organized around documents mostly. This is due to the fact that identifying entities (e.g., concepts) in specific collections such as scientific articles is more difficult than, say, in online news articles due to the *novelty* (i.e., new terms may be used which have not been previously observed in any other document/dictionary) and *specificity* (i.e., highly technical and detailed formalisms mixed with narrative examples) of the content.

While retrieving documents in an entity-centric fashion would also be beneficial for specialized domains, the difficulty of correctly extracting highly-specialized entities as well as the scarcity of semi-structured information available for specific documents are precluding such advances. As an example, the ScieceWISE portal[1] [3] is an ontology-based system for bookmarking and recommending papers for physicists. ScieceWISE is entity-centric, yet it requires human intervention to correctly extract the scientific concepts appearing in each new paper uploaded onto the system.

In this chapter, we tackle the problem of NER in highly-specialized domains such as scientific disciplines. We develop new techniques to identify relevant entities appearing in a scientific document, based on a set of features including n-gram statistics, syntactic part-of-speech patterns, and semantic techniques based on the use of external knowledge bases. In addition, we effectively combine our various features using a state-of-the-art machine learning approach in order to get the most out of our different families of features. The results of our NER approach can then be used for many applications, including to organize data on search engine results pages, to summarize scientific documents, or to provide faceted-search capabilities for

[1] http://www.sciencewise.info

literature search.

We experimentally evaluate the effectiveness of our methods over two manually labeled collections of scientific documents: a collection of papers from SIGIR 2012 conference (a well-known scientific conference on Information Retrieval), and a sample of research papers retrieved from arXiv.org. Our experimental results show how semantic-aware features overcome simple text-based features and how a combination of our proposed features can reach up to 85% overall Accuracy, significantly improving over state-of-the-art domain-specific supervised approaches based on maximum entropy [39]. In summary, the main contributions of this work are as follows:

- We tackle the problem of NER in the challenging context of idiosyncratic collections such as scientific articles.
- We describe a new, multi-step candidate selection process for named entities favoring recall (as standard techniques perform poorly in our context) and based on co-location statistics.
- We propose novel NER techniques based on semantic relations between entities as found in domain-specific or generic third-party knowledge bases.
- We extensively evaluate our approach over two different test collections covering different scientific domains and compare it against state-of-the-art NER approaches.
- We identify an effective combination of both syntactic and semantic features using decision trees and apply them on our collections, obtaining up to 85% Accuracy.

The rest of the chapter is organized as follows: We start with an overview of related work in the areas of named-entity recognition, keyphrase extraction, and concept extraction below in Section 3.2. We describe our overall system architecture and its main features (including PDF extraction, n-gram lemmatization, part-of-speech tagging, external knowledge bases, and n-gram ranking) in Section 3.3. Section 3.4 provides definitions of our ranking features. Section 3.5 describes our experimental setting and presents the results of a series of experiments comparing different combinations of features. Finally, we conclude and discuss future work in Section 3.6.

3.2 Related Work

Named entity recognition (NER) designates the task of correctly identifying words or phrases in textual documents that express names of entities such as people, organizations, locations, etc. During the last decades, NER has been widely studied and the best NER approaches nowadays produce near-human recognition accuracy for generic domains such as news articles. Several prominent NER systems use either hand-coded rules or supervised learning methods such as

maximum entropy [19] and conditional random fields [40]. These methods heavily rely on large corpora of hand-labeled training data, which are generally-speaking hard to produce. Besides the high costs attributed to the manual annotation of the training data, this also raises the problem of domain-specificity; For instance, models trained for news articles are most likely to perform well on such documents only [108].

In that context, there has been a lot of attention given to NER applied to newswire text (mostly because of the high quality of such texts), focusing on entity types such as locations, people, and company names. On the other hand, the task of NER for more domain-specific collections, e.g., for scientific or technical collections, remains largely unexplored, with a few exceptions including the biomedical domain where previous work has focused on specific entity types like genes, protein and drug names [130, 39]. In this chapter we focus on semantic-based NER over such domain-specific collections.

Open Information Extraction To address some of the above issues, researchers have recently focused on Web-scale NER (also known as Open Information Extraction) using automatic generation of training data [150], unsupervised NER based on external resources such as Wikipedia and Web n-gram corpora [80], and robust NER performance analysis across domains [118]. In this area, information extraction at scale is run over the Web to find entities and factual information to be represented in structured form [152, 31]. Instead, we focus on well-curated and highly-technical textual content. Compared to previous work in NER, we focus on effectively performing NER on domain-specific collections like technical articles. We apply state-of-the-art techniques together with specific approaches for scientific documents including the use of domain-specific knowledge bases to improve the quality of NER at a level comparable to the one achieved in news documents.

Key Term Extraction Another task related to this work is *key term extraction*. Key term extraction deals with the extraction and ranking of the most important phrases in a text. This can be used, for instance, in text summarization or tagging [16]. In [67], authors address this task as a ranking problem rather than a classification task. Contrary to NER research, many approaches in the area of key term extraction deal with technical and scientific document collections. Some recent evaluation competitions such as [73] are specifically geared towards scientific articles. Although the Precision of the top-performing systems is typically around 40% for such competitions, these results can be considered as rather high due to the specificity of the terms appearing in the scientific documents and the rather subjective nature of the ground-truth in that context. At this point, we want to emphasize that key phrases extraction is different from the task we address here, which aims at identifying *all* relevant entities in a document to enable further entity-centric processes (e.g., in the search engine).

The candidate identification step of term extraction systems typically filters all of the possible n-grams from the documents by frequency, retaining high frequency n-grams only. Some

methods use hand-coded part-of-speech tag patterns to provide additional filtering [142, 42], though hand-coded tag patterns are not always able to capture the variety of all valid entities due to tagging ambiguity (i.e., the same term may be considered either as a verb or as an adjective depending on the context). Instead, in our work we use standard frequency filtering with a re-weighting step to identify as many candidates as possible and part-of-speech tags as a feature to boost both Precision and Recall of NER.

The majority of keyphrase extraction studies use supervised models, the most commonly used approaches being naive Bayes [142, 41], decision trees [142] and support vector machines [76]. In our work, we use a decision tree-based classifier since it is able to handle easily both numerical and categorical data with little data preprocessing. Decision trees are also simple to interpret by the end-users who are the authors of scientific papers. Specifically, we base our work on a decision tree model and ensemble methods for feature selection using extremely randomized trees [46].

We also note that the work we present here actually lies in between the NER and key term extraction tasks. In standard NER, the goal is to identify all named entities mentioned in a document while in key term extraction the goal is to identify the most representative terms in a document. The task we address in this chapter is rather to identify the subset of named entities that are *valid* for the given idiosyncratic documents considered (see Section 3.3.1 for details).

Entity Linking Some previous work successfully used Wikipedia or DBpedia to identify significant terms in textual documents [92, 89, 32]. However, such methods operate only on the entities that already exist in the knowledge bases. The task of identifying entity mentions given a background corpus of entities is also known as *Entity Linking*. On the other hand, our goal is to also discover *new* entities from scientific documents, potentially by leveraging generic-purpose or specific knowledge bases.

3.3 System Overview

3.3.1 Problem Definition

The task we address is the identification of all valid entities related to a given domain in a domain-specific collection. In the context of this work, we define a *valid entity* as an n-gram representing a relevant concept of a scientific domain and not just as any real-world object. To give a clearer understanding of what a valid entity is in our case, let us look at a few examples. Consider the n-gram “Saving Private Ryan”. Usually such a string represents a valid entity referring to a popular movie, but it does not make much sense to mark this n-gram as valid in an Information Retrieval paper, where it was given as a query example. Another example illustrating the complexity of our task comes from disambiguation decisions. Consider the n-gram “large numbers”; It can be a valid entity in document is talking about *large numbers* in

a pure mathematical sense, but in many other cases it is just a linguistic construction.

To assess the performance of our approach, we use a standard set of evaluation metrics: Precision, Recall, F1 score, and Accuracy, which are computed on a per document basis (i.e., each item in our test collection is represented by a pair *(document, n-gram)*). These metrics allow us to show how well an approach performs both on true positives and true negatives and to discuss the resulting trade-offs.

In this work, we exclusively focus on the identification of n-grams entities with $n > 1$ because of the high level of inherent ambiguity that unigrams have in scientific literature. Many unigrams are ambiguous and can often be used both as entities and non-entities, even when inspecting a single document. Moreover, we argue that most unigram entities are very generic and can be recognized by simple dictionary lookups[2]. Thus, techniques like *entity linking* [92, 96] are in our opinion more suitable to address unigram entity recognition.

3.3.2 Framework

To evaluate our proposed approach, we have built a system that takes as input a set of scientific documents in PDF format and returns as output the set of n-grams appearing in the text of the documents that represent scientific concepts. Figure 3.1 below gives an overview of the architecture of our system.

The first components in our pipeline extract text from the input documents and perform some automatic preprocessing (e.g., lemmatization). The following steps consist in identifying the candidate entities that are potentially relevant concepts. The candidate selection step focuses on high Recall while keeping the number of candidate n-grams orders of magnitude lower than the total number of n-grams in a document. Finally, we use a series of approaches to select the valid n-grams among the candidates (focusing on high Precision). We discuss this pipeline in more detail in the following.

3.3.3 Data Preprocessing

Our system receives PDF documents as input and transforms them into raw text using an open-source library[3]. We then perform a series of preprocessing steps; First, we lowercase all words (except acronyms) appearing at the beginning of sentences to prevent duplicate entity creation in the latter steps. At this point, we make a separate copy of the resulting text (before lemmatization) on which we apply Part-Of-Speech (POS) tagging.

The first copy of the text is then lemmatized, using the a lemmatization approach based on the WordNet ontology [91]. We have opted for lemmatization in our context since the other

[2] an analysis of the tags verified by the users of the ScienceWISE platform shows that 23% of the tags are unigrams, and only 5% of them (1% overall) are not found in Wikipedia.

[3] We use Apache Tika http://tika.apache.org/ for this task.

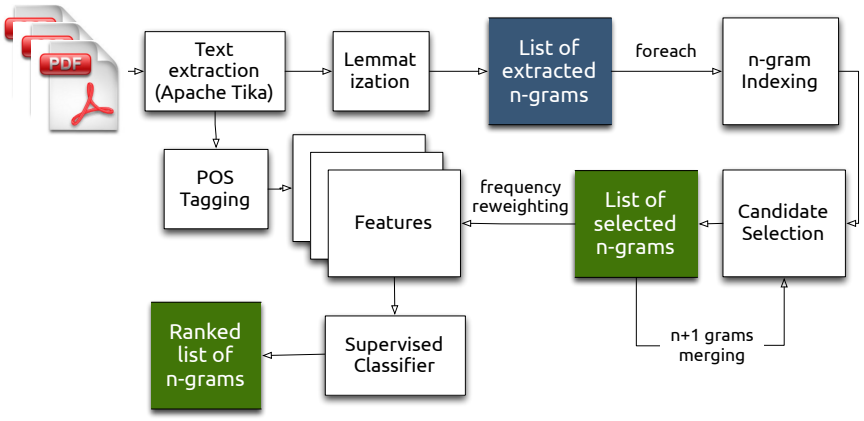

Figure 3.1 – Processing pipeline. First, the plain text is extracted from the PDF documents. Then, the text is pre-processed using lemmatization and POS tagging. Candidate n-grams are generated and indexed. Then, n-grams are selected based on a predefined set of features (see Section 3.3.4). Finally, a supervised approach (e.g., decision trees) is responsible to generate a ranked list of n-grams that have been classified as valid entities in the documents.

typical possibility, stemming, is too aggressive on scientific documents as it often conflates scientific concepts which should be kept distinct[4]. In the final step, we build an *n-gram index* from the resulting text to efficiently perform the candidate selection phase described below.

3.3.4 Candidate Selection

The goal of the candidate selection step is to extract as many candidate entities as possible from the scientific articles, while limiting the number of false positives. To achieve this goal, we extend techniques based on word co-locations [85]. First, we extract from the n-gram index all *bigrams* having a frequency (i.e., number of occurrences in the input document) greater than a threshold k (e.g., $k = 2$). Next, the extracted bigrams are joined together into trigrams; Two bigrams are joined if and only if it is possible to merge them to form a valid trigram (i.e., if the same word ends a bigram and starts another one). The resulting trigram frequency is then looked-up from the n-gram index.

This process is repeated for trigrams, up to the maximal n-gram size N considered ($N = 5$ in our experiments as for $N > 5$ we could not identify valid concepts in our test collections). The difference between simply restricting the frequency of any n-gram to k and our approach is that we can extract n-grams with a frequency lower than k: As can be seen on the graph of n-gram occurrence distribution depicted in Figure 3.2), there are many valid n-grams in the collection that appear just once or twice in the text, and removing them with a frequency threshold would result in a sharp decrease in Recall. Hence, after processing every document, we regroup the extracted n-grams from the entire collection and look them up again in every

[4] We have also performed our extraction experiments without any lemmatization and found that this reduces Recall by 4%.

document. This process preserves n-grams that passed the frequency threshold k in some papers, but not in others.

This collection-wide n-gram selection approach results in an increase of Recall from 42.2% to 96.1%. Alternatively, we also tried two further approaches: using the collection-level n-gram frequencies to serve as a cutoff frequency k, and running the n-gram merging process from scratch after adding collection-wide n-grams. These approaches yielded Recall values of 87.4% and 93.2% respectively.

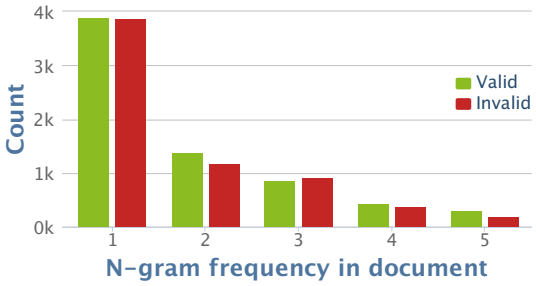

Figure 3.2 – Valid/invalid n-gram count distribution for the SIGIR collection. Only the first 5 frequencies are shown.

Removing Incomplete N-Grams In the last step, we apply a frequency reweighing process that takes into account the fact that some n-grams appear as part of other n-grams. We illustrate our reweighing mechanism by an example. Assume that in a document two bigrams "latent dirichlet" and "dirichlet allocation" appear both with frequency f, and that a trigram "latent dirichlet allocation" also appears with the same frequency. It is safe to say in that case that those two bigrams do not appear in the text as separate entities, but only as part of a bigger trigram. Our process hence starts from the longest n-grams (i.e., from n-grams with larger n), and proportionally decrements the frequency of the shorter n-grams that are subsumed by it. At the end of this process, we eliminate all n-grams having a re-calculated frequency equal to zero.

3.3.5 Supervised N-Gram Selection

Rather than simply weighting different features in order to determine whether an n-gram represents a correct concept or not, we apply machine learning approaches to learn to identify correct entities. We construct a feature space consisting of the features presented in Section 3.4 and use a manually labeled set of entities appearing in scientific documents as training data for our classifier. For classification, we used a method based on ensembles of Decision Trees [46], as it is one of the most robust and state-of-the-art machine learning approaches. Once trained, the classifier is then able to take as input a new document and—thanks to the processing pipeline depicted in Figure 3.1—to effectively select all valid scientific concepts from the document.

3.4 Features for NER

In this section, we describe the five different families of features used by our system to detect named entities in scientific documents. We propose different families of features ranging from simple syntactic POS patterns to features using third-party resources such as external knowledge bases and structured repositories like DBLP[5]. We also propose to combine our features using machine learning approaches. More specifically, we use decision trees to decide which n-grams correspond to valid concepts in the documents. This also allows us to understand which features are the most valuable in our context based on a hierarchy generated by our learning component.

While having many features to predict the correctness of an n-gram as a scientific concept in a paper is positive, there is a risk of overfitting: The learned model might be to much customized to the training data and its features thus creating a model which is not generalizable. To avoid such problem we define families of features which are then evaluated by means of ablation experiments.

3.4.1 Part-of-Speech Tags

Part-Of-Speech (POS) tags have often been considered as an important discriminative feature for term identification. Many works on key term identification apply either fixed or regular expression POS tag patterns to improve their effectiveness. Nonetheless, POS tags alone cannot produce high-quality results. As can be seen from the overall POS tag distribution graph extracted from one of our collections (see Figure 3.3), many of the most frequent tag patterns (e.g., *JJ NN* tagging adjectives and nouns[6]) are far from yielding perfect results.

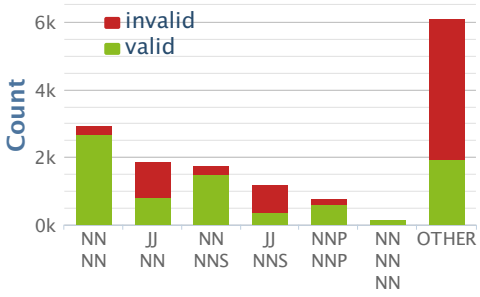

Figure 3.3 – Top 6 most frequent part-of-speech tag patterns of the SIGIR collection, where *JJ* stands for adjectives, *NN* and *NNS* for singular and plural nouns, and *NNP* for proper nouns.

Given those results, we designed several features based on POS tags that might perform better than predefined POS patterns. First, we consider raw POS tags where each POS tag pattern

[5] http://dblp.dagstuhl.de/

[6] see http://www.cis.upenn.edu/~treebank/ for an explanation on POS tags

represents a separate binary feature. Though raw POS tags can provide a good baseline in some settings, we do not expect them to perform well in our case because of the large variety of POS tag patterns in both collections, many of which can be overly specific.

A more appealing choice is to group (or *compress*) several related POS tag patterns into one aggregated pattern. We use two grouping techniques: Compressing all POS tag patterns by only taking into account i) the first or ii) the last POS tag in the pattern. Using the compressed POS tag versions, we significantly reduce the feature space, which is the key to achieve higher performance and allows for model generalization. We discuss those two schemes in more detail in Section 3.5.2. To perform POS tagging, we used a standard approach based on maximum entropy [121].

3.4.2 Near N-gram Punctuation

Another potentially interesting set of features closely related to POS tags is punctuation. Punctuation marks can provide important linguistic information about the n-grams without resorting to any deep syntactic analysis of the phrase structure. For example, the n-gram *"new summarization approach based"*, which does not represent any valid entity, has a very low probability of being followed by a dot or comma, while the n-gram *"automatic music genre classification"*, which is indeed a valid entity, often appears either at the beginning or at the end of a sentence.

The contingency tables given in Table 3.1 and Table 3.2 illustrate this: The *+punctuation* and *-punctuation* rows show, respectively, the counts of the n-grams that have at least one punctuation mark in any of its occurrences and the counts of the n-grams that have no punctuation mark in all their occurrences. From the tables, we observe that the presence of punctuation marks (+punctuation) either before or after an n-gram occurs twice as often for the n-grams that are valid entities compared to the invalid ones. We also observe that the absence of punctuation marks after an n-gram happens less frequently for the valid n-grams than for the invalid ones.

Table 3.1 – Contingency table for punctuation marks appearing *immediately before* the n-grams.

	Valid	**Invalid**	**Total**
+punctuation	1622	847	2469
−punctuation	6523	6065	12588
Totals	8145	6912	15057

Thus, both directly preceding and following punctuation marks are able to provide relevant information on the validity of the n-grams and can be used as binary features for NER.

Table 3.2 – Contingency table for punctuation marks appearing *immediately after* the n-grams.

	Valid	**Invalid**	**Total**
+punctuation	4887	2374	7261
−punctuation	3258	4538	7796
Totals	8145	6912	15057

3.4.3 Domain-Specific Knowledge Bases: DBLP Keywords and Physics Concepts

DBLP is a website that tracks and maintains bibliographic references for the majority of computer science journals and conference proceedings. The structured meta-data of its records include high quality keywords that authors assign to their papers.

Author-assigned keywords represent a very reliable source of named entities for documents related to this specific domain. In fact, the overall Precision of n-grams from author-assigned keywords for our computer science dataset is 95.5% (with 27.4% Recall), and hence can be used as a highly discriminative feature.

While DBLP provides high quality annotations for computer science documents, there is no such knowledge base for our physics collection. Thus, we decided to perform a similar matching using the concepts from one of the largest physics ontology available—the ScienceWISE ontology[7]. All the concepts in this ontology represent valid named entities which, as for DBLP, can be used as a highly discriminative feature.

3.4.4 Wikipedia/DBPedia Relation Graphs

Wikipedia is by far the largest general-purpose knowledge-base currently available. In the context of our task, Wikipedia exhibits the following valuable features[8]:

- The majority of pages in Wikipedia represent valid named entities.
- Pages are interconnected with each other through links appearing in the page body and through their categories.
- Many pages have alternative labels which are encoded by a special “redirects” property.

We base our Wikipedia features on collection statistics. Specifically, we use a machine-processable version of Wikipedia called DBPedia[9], which contains all entities in Wikipedia described in a structured format and interconnected to other datasets. We start by computing

[7] http://sciencewise.info/ontology/

[8] Every feature in the above list is freely accessible through the Wikipedia API at http://en.wikipedia.org/w/api.php.

[9] http://dbpedia.org/

the Precision and Recall values when matching Wikipedia pages with the n-grams from our collections. Table 3.3 shows the resulting values for two cases: i) exact string matching with page title and ii) matching allowing variants based on the "redirects" property. As expected, we observe that allowing flexible matching with redirects results in a significant growth in Recall, with some loss in Precision[10].

Table 3.3 – Precision/Recall values for Wikipedia features.

	SIGIR		**Physics**	
	Precision	**Recall**	**Precision**	**Recall**
String matching	0.9045	0.2394	0.7063	0.0155
Matching with redirects	0.8457	0.4229	0.7768	0.5843

Furthermore, taking into consideration the relatively low Precision of exact Wikipedia matchings, one can try to improve the above technique by finding further methods to separate the valid entities from the invalid ones. Hulpus *et al.* [64] recently observed that interlinked Wikipedia pages are much more likely to form a connected component in the Wikipedia category graph than random pages. Given that finding, we use the size of the connected component a Wikipedia page belongs to as an additional feature for valid concepts.

Following the approach in [64], we construct the neighboring page graph by following relationships in DBPedia of types {*broader,subject,related*} for up to two hops in both directions. The two hops threshold was chosen based on previous research from [64], which claimed that bigger distances result in much larger graphs and introduce noise. The Wikipedia administrative categories and pages referring to etymology (e.g., "English phrases") are excluded using an existing list of stop URIs[11].

Figure 3.4 shows how often the connected component of a given size contains more valid than invalid entities, while Figure 3.5 shows the average percentages of valid and invalid entities in a component of a given size. We observe that larger connected components tend indeed to contain more valid entities than smaller ones.

Based on the analysis made above, we construct the following set of NER features using relation graphs:

- *is wiki*: whether a candidate n-gram can be exactly matched to a Wikipedia page title,
- *is redirect*: whether a candidate n-gram can be matched using an alternative spelling of a Wikipedia page,
- *component size*: the size of the connected components an n-gram belongs to, constructed with and without the redirect property,

[10] Though Precision for the physics collection actually goes up, most likely because of the very low number of n-grams exactly matching—only about 60 cases.

[11] http://uimr.deri.ie/sites/StopUris

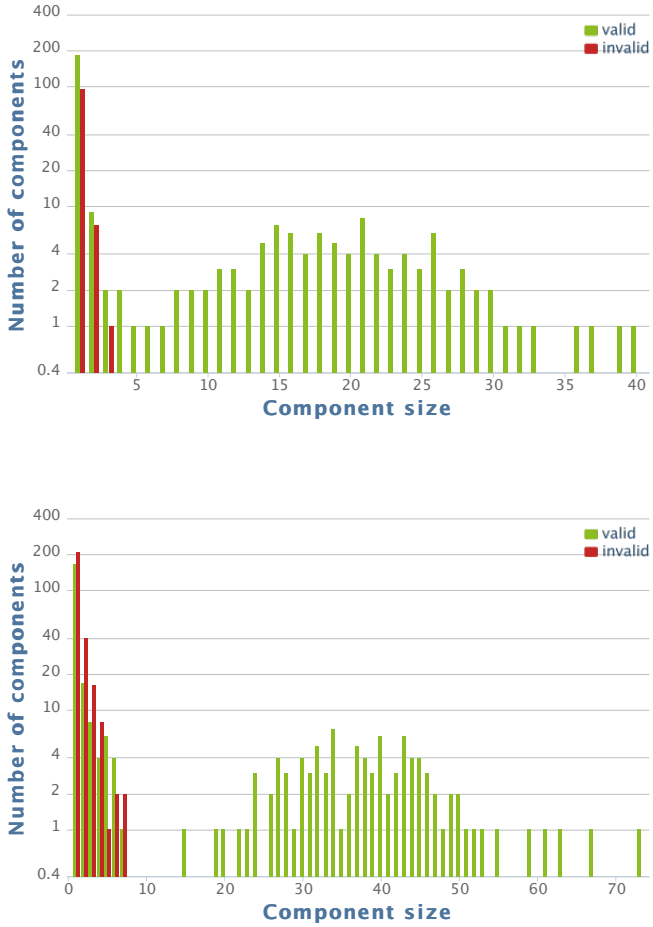

Figure 3.4 – DBPedia connected component sizes for valid/invalid n-grams without (top) and with (bottom) the use of Wikipedia's redirect property.

- *component+DBLP*: a binary feature, equals to 1 when an n-gram appears in the same connected component with at least one DBLP keyword, and to 0 otherwise;
- *wikilinks*: the number of outgoing links in the Wikipedia page body to other Wikipedia pages.

3.4.5 Syntactic Features

In addition to the features described above, we also test a series of more common syntactic features that are often used by other NER classifiers, including:

- the *n-gram length* in words,
- whether the n-gram is uppercased,

- the number of other n-grams the given n-gram is part of in the document.

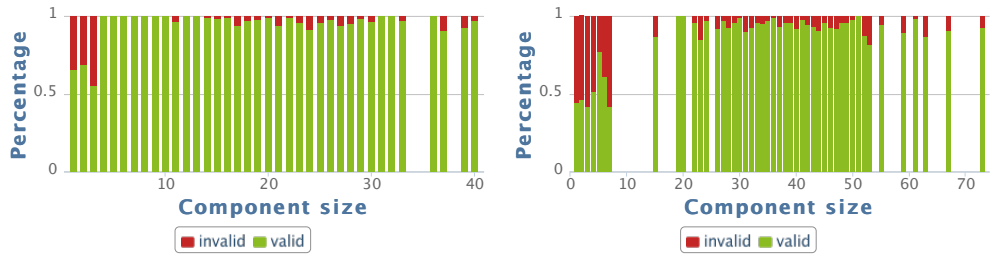

Figure 3.5 – DBPedia connected component size percentage distribution for valid/invalid n-grams without (left) and with (right) using Wikipedia redirect properties.

3.5 Experimental Evaluation

3.5.1 Experimental Setting

In this section, we empirically evaluate the NER techniques proposed above. We evaluate the quality of our features as well as how to best combine them over two distinct test collections for which ground truth entity annotations have been manually created by domain experts from two specific domains: Computer Science and Physics.

Dataset Description Our first dataset contains 100 randomly selected papers taken from the SIGIR 2012 conference proceedings, while our second dataset contains the same number of recent (2012) articles taken from the High Energy Physics (hep-ph) section from the arXiv.org pre-print repository.

Our system extracted 21,531 candidate n-grams in total from the first dataset, of which 8,814 n-grams were unique. Overall, 15,057 n-grams were judged, of which 8,145 were labeled as valid and 6,912 as invalid.

In the second dataset, our system extracted 18,129 candidate n-grams, of which 7,880 n-grams were unique. Overall, 11,421 n-grams were judged, of which 5,747 were labeled as valid and 5,674 as invalid[12].

The judgments were performed on a per-document basis, meaning that an n-gram was considered as a relevant scientific concept if it represented a valid entity in the scope of a particular document from the collection. Thus, each judgment in the collection is connected

[12] Both datasets and ground truth data are made available for online exploration and download at https://github.com/XI-lab/scientific_NER_dataset.

to the source document ID (document title for the first collection and arXiv.org ID for the second). All judgments have been made by one or more experts from the given scientific field.

Relevance Judgments Deciding whether or not a given n-gram represents a valid scientific entity can be subject to discussion. Therefore, the guidelines we have given to the assessors stipulate that an n-gram should be considered as a valid entity if it belongs to the domain of the document and satisfies any one of the two following conditions:

- it would make sense to take the n-gram and create a thesaurus/encyclopedia entry about it, or
- the n-gram could be used by an expert to search/filter the papers according to domain-specific (e.g., scientific or technical) criteria.

3.5.2 Experimental Results

Individual Features Table 3.4 presents the effectiveness of our individual feature families over the Physics test collection, while Table 3.5 presents similar results for the SIGIR collection. We observe that well-performing features on the Physics collection are based on POS tags or on the connected components obtained from *redirect* information in Wikipedia. We also evaluate our set of basic syntactic features (see Section 3.4.5) for comparison. On the SIGIR collection, we observe that the best performing features are based on POS tags both in terms of F1 and Accuracy. In terms of Precision, the best approach is the one using the graph connected components.

Table 3.4 – Empirical results for individual feature families on the Physics collection.

	Precision	**Recall**	**F1 score**	**Accuracy**
Compressed POS tags	0.5742	0.9511	**0.7160**	0.6198
Component	0.5039	**1.0**	0.6702	0.5039
Component+Redirects	**0.8116**	0.5572	0.6605	**0.7117**
Punctuation	0.5039	**1.0**	0.6702	0.5039
Syntactic	0.5940	0.1771	0.2728	0.5243

Table 3.5 – Evaluation results for individual feature families on the SIGIR collection.

	Precision	**Recall**	**F1 score**	**Accuracy**
Compressed POS tags	0.8183	0.7307	**0.7715**	**0.7772**
Component	**0.8981**	0.2280	0.3635	0.5888
Component+Redirects	0.8883	0.3869	0.5388	0.6588
Punctuation	0.6414	**0.9450**	0.7642	0.6820
Syntactic	0.6819	0.2124	0.3236	0.5429

Feature Comparison To find effective feature combinations, we use a decision tree ensemble classifier with default parameters from the scikit-learn library[13] [105]. To prevent the classifier from over-fitting the training data, we restrict the minimum number of samples in the leaves to 100 and the maximum depth of the tree to 5. All the results presented below are the mean values resulting from a 10-fold cross-validation of our supervised approach.

We compare the effectiveness between pairs of competing features: compressed and uncompressed POS tags on one hand (see Section 3.4.1), and building DBPedia connected components with and without the "redirects" property on the other hand (see Section 3.4.4).

Tables 3.6 and 3.7 show the Precision, Recall, F1, and Accuracy values over both collections for different combinations of compressed and uncompressed POS tags features and DBPedia category graph features with and without the *redirect* property. We observe that adding Wikipedia redirects allows to significantly improve Recall in most cases without a significant loss in Precision. Improved Recall is somewhat expected since the *redirect* property allows to match many more Wikipedia concepts. More importantly and as mentioned earlier, this Recall growth does not produce any major loss in Precision, which results in a consistent growth in Accuracy.

Another important result here is that compressed POS tags produce roughly the same Precision values as uncompressed ones with a much smaller number of features. The reason is that the uncompressed POS tag pattern space is much richer than the one of the compressed patterns, which in theory could allow classifiers to yield better performance at the price of possible over-fitting. However, by using a smaller feature space we observe a minor decrease in Precision on both collections with a higher F1 score on the SIGIR collection. Hence, we conclude that compressing POS tags is a better choice since it allows for better model generalization.

Feature Selection Table 3.8 shows the NER features we propose ranked by the score they yield when combined using randomized trees as suggested by [46] on the SIGIR collection. As we can see, the simple techniques based on POS patterns is highly discriminative. However, POS tags are by themselves not sufficient; Other top features include the ones that look at external knowledge bases such as DBLP and the structure connecting the DBPedia entities mentioned in the document.

Table 3.9 shows the feature ranking based on randomized trees for the Physics collection. In this case, we observe that the most indicative features are the ones based on external ontologies and knowledge bases. In this case, we believe that such features are most distinctive due to the highly technical terms used in Physics and due to the somewhat slower churn of new terminology as compared to the IR field, which is a much younger research area.

In conclusion, we observe that the use of domain-specific knowledge-bases is an effective feature for NER on technical collections.

[13] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

Table 3.6 – Evaluation results for different feature combinations on the SIGIR collection. The symbols * and $^{*}*$ indicate a statistical significant difference (t-test $p < 0.05$ and $p < 0.01$ respectively) as compared to the best approach (in bold font).

All features	**Precision**	**Recall**	**F1 score**	**Accuracy**	**N features**
+ Uncompressed POS + Component	**0.8794**	0.8058**	0.8409**	0.8429*	54
+ Compressed POS + Component	0.8475**	0.8524**	0.8499**	0.8448**	9
+ Uncompressed POS + Component+Redirects	0.8678**	0.8305**	0.8487*	0.8473	50
+ Compressed POS + Component+Redirects	0.8406**	**0.8769**	**0.8584**	**0.8509**	7

Table 3.7 – Evaluation results for different feature combinations on the Physics collection. The symbol * and ** indicate a statistically significant difference (t-test $p < 0.05$ and $p < 0.01$ respectively) as compared to the best approach (in bold).

All features	**Precision**	**Recall**	**F1 score**	**Accuracy**	**N features**
+ uncompressed POS + Component	0.8253*	0.6567*	0.7311**	0.7567	53
+ compressed POS + Component	0.7941**	**0.6781**	0.7315**	0.7492**	4
+ uncompressed POS + Component+Redirects	0.8339	0.6674*	**0.7412**	**0.7653**	50
+ compressed POS + Component+Redirects	**0.8375**	0.6479**	0.7305**	0.7592*	6

Table 3.8 – Ranked list of feature importance scores on the SIGIR collection. Selected number of features: 7

Feature name	Importance score
NN STARTS	0.3091
DBLP	0.1442
Component+DBLP	0.1125
Component	0.0798
VB ENDS	0.0386
NN ENDS	0.038
JJ STARTS	0.0364

Table 3.9 – Ranked list of feature importance scores on the Physics collection. Selected number of features: 6

Feature name	Importance score
ScienceWISE	0.2870
Component+ScienceWISE	0.1948
Wikipedia Redirect	0.1104
Component	0.1093
Wikilinks	0.0439
Participation count	0.0370

Feature Ablation Analysis Finally, we evaluate the contribution of the individual features to the overall feature combination by a hold-out experiment: We learn a new model by removing each time a feature family to measure the impact of that feature on the overall best possible combination of the features (85% Accuracy on SIGIR and 77% Accuracy on Physics).

Table 3.10 shows the effectiveness obtained by discarding one feature family for the Physics collection. As we can see, the highest loss in effectiveness (-24% F1 score) is observed when removing the background ontology of scientific terms. For the SIGIR collection (see Table 3.11), we observe that the biggest loss is due to the removal of POS tags (-19% F1 score) confirming the results of feature selection based on randomized trees.

Generally speaking, we see the importance of using domain-specific knowledge bases as well as linguistic properties.

Maximum Entropy Classifier Baseline As a method to compare to, we chose the state-of-the-art Maximum Entropy Classifier (MaxEnt) for Named Entity Recognition [15].

In contrast to our approach depicted in Figure 3.1, this classifier receives the *full text* of the document extracted from the PDF file together with a training set of manually labeled scientific concepts appearing in it. After training the model, the classifier is able to detect

Table 3.10 – Effectiveness values for different feature combinations on the Physics collection. The symbols * and ** indicate a statistically significant difference (t-test $p < 0.05$ and $p < 0.01$ respectively) as compared to the approach using all features.

Feature set	Precision	Recall	F1 score	Accuracy
All Features	0.8375	0.6479	0.7305	0.7592
−ScienceWISE (SW)	0.7861**	0.6072**	0.6850**	0.7187**
−Component+SW	0.8375	0.6479	0.7305	0.7592
−Wikipedia Redirect	0.8368	0.6483	0.7305	0.7590
−Component	0.8354	0.6391	0.7241*	0.7547

Table 3.11 – Effectiveness values for different feature combinations on the SIGIR collection. All differences with respect to the use of all features are statistically significant with $p < 0.01$.

Feature set	Precision	Recall	F1 score	Accuracy
All Features	0.8406	0.8769	0.8584	0.8509
−POS tags	0.9186	0.5370	0.6776	0.7368
−DBLP	0.8330	0.8397	0.8362	0.8305
−Component+DBLP	0.8181	0.8855	0.8505	0.8395
−Component	0.8212	0.8739	0.8467	0.8369

unseen scientific concepts given the full text of a new document.

To evaluate the MaxEnt NER approach, we trained it on 80% of SIGIR data and used the rest 20% as a test dataset[14].

During the experiment, 3,380 new n-grams were extracted, out of which 346 new valid entities were discovered. All n-grams extracted during the comparison experiment were fully judged and added to the evaluation datasets described previously.

Table 3.12 – Evaluation results for maximum entropy classifier on SIGIR collection.

	Precision	Recall	F1 score
MaxEnt NER Baseline	0.6566	0.7196	0.6867
Our Approach (using Decision Trees)	0.8121	0.8742	0.8420

For a fair comparison, we evaluate our top-performing supervised method on the same data. The results of this experiment are presented in Table 3.12. As can be observed, the decision tree-based method outperforms the state-of-the-art MaxEnt approach by roughly 15% both in Precision and Recall [15].

[14] The parameters of the tagger were estimated using the *generalized iterative scaling* method [30].

[15] Accuracy score is not shown in the table since the notion of *true negative* is not valid for the MaxEnt method, where literally every non-positive n-gram can be considered as negative.

3.5.3 Results Discussion

Based on the experimental results described above, we first observe that the NER approach for idiosyncratic collections we propose here substantially outperforms state-of-the-art supervised NER approaches such as MaxEnt. As an example, our best supervised approach yields a F1 score of 84% on the SIGIR collections, compared to 69% for MaxEnt.

We also note that the most effective features among the ones we propose vary depending on the test collection. However, we observe that both the feature family based on the entity-graph structure and the family based on external domain-specific knowledge bases are key to enhance NER effectiveness for idiosyncratic collections.

Finally, while comparing the two test collections, we note that the Physics collection lead to overall lower effectiveness scores. This may be explained by the more formal terminology used in that scientific domain, which makes the identification of valid scientific concepts more challenging as compared to Computer Science academic documents.

3.6 Conclusions

Being able to identify entities in textual documents is known to be beneficial for many tasks, including document search, integration, classification, or summarization. While supervised methods are often used for NER in Web documents such as news articles, novel approaches are needed to perform NER over more specific domains such as for scientific papers.

In this chapter, we addressed the task of NER for domain-specific collections by taking advantages of n-gram-based features. We proposed and experimentally validated over two different test collections novel NER features and their combinations using machine learning classifiers trained over data created by domain experts. More specifically, our novel features for domain-specific NER include the analysis of entity-graph components as well as the use of external domain-dependent knowledge bases such as DBLP for Computer Science or the ScienceWISE ontology for Physics.

Our results show that the analysis of entity-graph structures and the use of external knowledge bases yield significantly better results in our context. For the two collections we considered, the best performance was obtained by our combined method, yielding up to 85% Accuracy.

Further improvements could be obtained by enhancing other components of our system pipeline. For example, advanced PDF extraction approaches could be used to detect bibliographic sections, or to identify titles and emphasized text, which may both allow to improve candidate selection and construct new feature sets. Such approaches providing more structured input would probably yield higher effectiveness values for the task we consider.

As a possible extension of our approach, one could use additional components in the processing pipeline. For example, entity linking approaches allowing to disambiguate entities identi-

fied in the text could be exploited. In this work, we directly matched n-grams to Wikipedia entries, though it might be more effective to perform disambiguation first. In the next chapter, we investigate entity disambiguation problem for scientific literature domain, where we propose a novel approach based on contextual entities and ontological connections among them and evaluate it in comparison to several traditional disambiguation methods.

4 Ontology-based entity disambiguation

4.1 Introduction

The number of scientific papers getting published is rapidly increasing. To support the discovery of new scientific results as well as exploratory endeavors within a new field of interest, modern systems rely on annotated collections of scientific papers. One example of such systems is PubMed, which uses the MeSH taxonomy[1] to annotate topics of scientific papers and to enable search over annotations. Annotations are usually created manually by the authors when creating or publishing a new document but can also, in some cases, be generated automatically, especially when performed at word-level. One example of such an automatic annotation system for scientific papers is ScienceWISE[2], which automatically annotates papers from the physics domain by adopting an expert-curated ontology as background information. Another example is Utopia[3], a system integrating visualization and data-analysis features that has been used by the editors of the Biochemical Journal (BJ) in a successful pilot.

In automatic annotation systems, most annotation errors are originating from ambiguous terms, which may lead to the wrong concepts being identified. One example in the Physics domain is the term 'cluster' which may refer to a 'cluster of galaxies' or to a 'cluster of stars', which are two very different concepts. Usually, the correct sense can be identified by the reader and by automated approaches using the context (e.g., the topic of the paper). In other cases, it might be necessary to take into account some particular background knowledge related to the specific research topic addressed in the paper. While an expert in the field might be able to determine the correct disambiguation in a scientific article thanks to his professional background, automatic approaches often fail to disambiguate the terms correctly without such knowledge.

For this reason, we propose a semi-supervised method for Entity Disambiguation (ED) for the scientific literature domain. The task we address is the disambiguation of scientific

[1] http://www.nlm.nih.gov/mesh/
[2] http://sciencewise.info
[3] http://getutopia.com/

terms and acronyms used in scientific abstracts. Our approach is based on the use of both contextual information from the document as well as a background knowledge-graph built and maintained by the scientific community. While no manually created annotated data is necessary to train our models, the proposed approach is semi-supervised in the sense that it exploits existing relations among entities in a background ontology that can be either manually or automatically generated.

We experimentally evaluate our approach over two different test collections, one based on the ScienceWISE Web portal used to semantically annotate, bookmark, and share papers in the Physics domain, and one based on the MeSH index for the MEDLINE corpus in the biomedical domain [69].

The main contributions of this work are:

- the definition of a ED task for a collection of scientific abstracts that are semantically annotated via a background ontology;
- novel efficient and effective approaches to ED that exploit both collection statistics as well as entity relations in the background ontology graph;
- a new test collection for ED over a background ontology graph and its entity relations;
- an experimental comparison of the proposed approach against prior disambiguation approaches over two different test collections, showing that our ontology-based methods are both more effective and more efficient than state-of-the-art approaches based on context vectors or automated classification when relying on a high-quality ontology.

The rest of this chapter is organized as follows: Section 4.2 describes previous work in the area of Entity and Word Sense Disambiguation (WSD). We define the problem of ED for semantically annotated scientific papers and propose a new approach leveraging collection statistics and relations among existing concepts in Section 4.3. Section 4.4 describes our experimental setting and presents the results of a series of experiments comparing our approach to existing ED methods. Finally, we conclude in Section 4.5.

4.2 Related Work

In this chapter, we target the scenario of Entity Disambiguation for scientific document collections. This is a compelling research topic, especially when considered in the context of online digital libraries offering metadata about scientific publications, like Bibsonomy [62] or ScienceWISE [3]. In the context of this work, we consider the problem of Entity Disambiguation as a sub-class of the Word Sense Disambiguation (WSD) problem, where instead of choosing the right sense for a single word, we need to choose the right entity for an entity mention.

The general problem of WSD has been widely studied in the past (see [99] for a survey). Both supervised and unsupervised approaches to WSD have been proposed.

Supervised approaches consider an initial set of training examples over which a model to disambiguate terms in documents is learned. A popular approach is Naïve Bayes [20], which is known to be effective but not particularly efficient. Other more efficient supervised methods based on Support Vector Machines have been proposed as well [79]. In this work, we propose a semi-supervised method that does not require training evidence but that is based on existing relations among domain entities within a manually curated ontology graph. We also compare our method against k-Nearest Neighbors (kNN), which is one of the most effective supervised approaches to WSD [28].

Knowledge-based methods are directly related to the approach we propose in this work. Knowledge-based methods adopt background information to select the correct sense of a term in a document. The most popular resource used by such approaches is WordNet [38], a machine-readable lexicon of word senses and linguistic relations. While very useful, the disadvantage of such a general-purpose resource lie in its lack of domain-specific information. In a recent paper [100], Navigli *et al.* propose a similar approach to supervised WSD based on text classification that also exploits a WordNet graph as background information. Our approach is different in a way that it is able to exploit *domain-specific* rather than general-purpose ontologies and does not require any training.

Another approach which, similarly to ours, proposes a method that leverages both a background knowledge-base as well as corpus statistics is [68]. In that work, the authors propose the use of a machine-readable dictionary over which similarity values are computed and used for clustering terms. On the other hand, our work aims at analyzing semantic relations among terms in the ontology in order to understand the intended meaning of a term. Our experiments also show higher accuracy values as compared to [68].

Standard test collections exist to evaluate and compare WSD approaches. In this work, we use an existing collection for WSD in the context of scientific documents which is based on the MeSH vocabulary [69]. Additionally, we create a novel test collection specifically targeting the scenario where a community-maintained background ontology as well as expert generated ground truth annotations are available.

4.3 Graph-Based Disambiguation Models

4.3.1 Ontology-based ED: Task Definition

The task we are focusing on here is Entity Disambiguation given a domain-specific ontology $O = \{E, R\}$ containing entities E and relations R among them. In the context of this chapter, we use "term" to denote a single word (separated by white spaces or any other punctuation symbol) and "entity" to denote the set of n-grams ($n \geq 1$) that define all possible labels of an

Table 4.1 – Examples of ECVs (main form) from the ScienceWISE collection.

Entity	ECV(Entity)
Star formation efficiency	(Instability, 4), (Supernova, 2), (Milky Way, 3), ...
Support vector machine	(Bayesian, 1), (Neural network, 2), (Classification, 11), ...
Markov decision process	(Probability , 10), (Reinforcement learning, 4), ...

entity in the ontology (e.g., "Milky Way Halo", "MW Halo"). We define the set $E_U \subset E$ as the set containing the n-grams that occur only once across the ontology (assuming that each entity in the ontology has at least one unique form, so-called *main form*) and a set $E_A \subset E$, which contains all the n-grams that occur more than once, such that $E_U \cup E_A = C$ and $E_U \cap E_A = \emptyset$.

The ontology is used to identify and extract entities from textual documents: given a document collection $D = \{d_1, .., d_n\}$, we extract from each document d_i a list of entities $e_1, .., e_n$ based on normalized n-gram matching. In some ambiguous cases, an extracted n-gram may refer to different entities in the ontology. In such cases, we define the ED task for an ambiguous n-gram as the selection of the right entity in the ontology among a list of candidate-matching entities. Details on the entity identification process are provided in Section 4.4.4.

4.3.2 Entity Context Vectors for Disambiguation

The first approach we adopt for ED over a scientific document collection is based on *context vectors*, which is a commonly used unsupervised approach for disambiguation (see, for example, [1]). A context vector $\vec{cv}(e_i)$ for an entity $e_i \in E_A$ is defined as $\vec{cv}(e_i) = \{(t_j, score_j) | t_j \in T\}$, where T is the space of all terms from the document collection. Such vectors may either contain binary values indicating whether t_j co-occurs or not in the same documents as c_i, or more informative values such as the frequency score of such co-occurrences.

Here, we define and use an extension of context vectors which—instead of using all words in the document context—first identifies entities in d based on the background ontology using entity linking methods (e.g., [56]). Thus, we define an *Entity Context Vector* (ECV) $\vec{ecv}(e_i)$ for an entity $e_i \in E_A$ as $\vec{ecv}(c_i) = \{(e_j, score_j) | e_j \in E_U\}$. The only difference with classic context vectors is that instead of considering all possible words in the textual context, we restrict our analysis on the co-occurrence of entities described in the ontology. An example of ECVs from our test collections is shown in Table 4.1.

Similarly, a *Document Entity Context Vector* (DECV) $\vec{decv}(d_i)$ is a vector consisting of all the entities identified in a document $d_i \in D$. Examples of DECVs are provided in Table 4.2. We define $\vec{decv}(d_i) = \{(e_j, score_j) | e_j \in E_U\}$. Once ECVs and DECVs have been constructed, it is possible to perform disambiguation by means of a similarity score between ECVs of the candidate matching entities and the DECV where the ambiguous entity mentions have been identified. In the experiments here, we rank candidate ECVs by cosine similarity scores with the target DECV.

Table 4.2 – Examples of DECV from the ScienceWISE collection.

DocID	DECV(DocID)
1	(Milky Way, 1), (Electron neutrino, 1), (Electron antineutrino, 1), ...
2	(Local analysis, 1), (Poynting-Robertson effect, 1), (White dwarf, 3), ...

4.3.3 Graph-based Approaches to WSD

Assuming that an ontology storing domain entities and their relations is available, it is possible to define advanced WSD methods that exploit such relations as well. Let us define a graph $O = \{E, R\}$ where nodes $e \in E$ are entities in the ontology and edges $r_l(e_i, e_j) \in R$ represent the labeled relations between different entities.

A first possible disambiguation method (minDist) that exploits such an additional structure is based on the distance between entities in the graph. Given an ambiguous n-gram and its candidate matching entities $CE = \{ce_1 \dots ce_n\}$ we select one entity based on the minimum distance with respect to all the other entities $DE = \{de_1 \dots de_n\}$ present in d, where the distance between two entities $dist(ce_i, de_j)$ is given by the shortest path connecting them in the O graph:

$$score(ce_i) = \min_{de_j \in DC} dist(ce_i, de_j) \tag{4.1}$$

A different approach (Ontology Shortest-Path, OSP) is also based on the ontology graph, but ranks candidate entities based on the average distance to all entities in d:

$$score(ce_i) = \frac{\sum_{de_j \in DE} dist(ce_i, de_j)}{|DE|} \tag{4.2}$$

Finally, the third approach (NN) we explore in this work is based on the neighborhood of the candidate matching entities given the ontology O. Thus, the confidence score to rank a candidate entity ce_i for a document d is given by the number of co-occurring neighbors e_j of ce_i in d:

$$score(ce_i) = |\{e_j | e_j \in EC \wedge dist(ce_i, e_j) = 1\}| \tag{4.3}$$

Those three techniques to score and rank candidate matching entities are experimentally compared over two different test collections in Section 4.4.

4.3.4 Combination of disambiguation approaches

The approaches for disambiguation described so far provide a score (e.g., similarity score between vectors) that indicates the confidence level of the disambiguation. Therefore, it is possible to combine different approaches together, for instance using a simple linear combina-

tion of their confidence scores and thus reach potentially better decisions based on multiple evidences. In this work, we propose and evaluate a mixture model among pairs of approaches A and B to circumvent the problem of having several parameters to learn at once:

$$score(cc_i) = \alpha score_A(cc_i) + (1-\alpha) score_B(cc_i), \alpha \in [0,1] \tag{4.4}$$

4.4 Experimental Evaluation

4.4.1 Experimental Setting

We evaluate the proposed models over two different test collections: one based on the MSH collection [69] and another one from the ScienceWISE system. Both collections contain a set of abstracts from scientific publications. In most cases, online digital libraries let guest users or crawlers only access the abstracts of the papers they store. Hence, we decided to restrict the document corpus to those abstracts only.

We consider four baseline approaches in the following. The first baseline approach we use for comparison is the random baseline (that is commonly used for comparison in WSD, see for instance [99]), which randomly assigns one among all the possible entities to the ambiguous entity mention. The more ambiguous the mention, the less effective this random baseline gets. Another simple baseline we consider is to always select the most frequent entity (as appearing in the document collection) among the candidate matching entities. We also compare the ECV-based approach against standard context vectors constructed over all the terms appearing in the document instead of only considering the extracted entities. The fourth baseline we use for comparison is the state-of-art supervised method based on Naïve Bayes (NB) classifier. We train it over 7'641 and 2'952 manually disambiguated documents for the MSH and ScienceWISE collection respectively.

The evaluation measures commonly used for WSD are Precision and Coverage (i.e., Recall). As our approaches always retrieve an entity for each ambiguous mention extracted from the abstracts, we only report Precision values in the following. To compare different approaches and to validate potential improvements, we measure statistical significance by means of a paired t-test considering a difference significant when $p < 0.05$. We describe the two document collections we used for our experiments below.

4.4.2 MSH Collection

The first document collection we use for evaluating our approaches consists of abstracts from the biomedical domain [69]. Each element of the test collection represents one ambiguous entity mention, its corresponding abstract and the correct entity among all the available entities. The test collection also contains all possible entities for each mention in a separate file.

To build the appropriate Entity Context Vectors for the MSH collection, we used the RESTful text annotator service offered by bioontology.org [4]. As a backend ontology for the annotation process, we used the Medical Subject Headings (MeSH) ontology[5], which is used by MEDLINE indexers to annotate the textual contents of biomedical articles. To focus exclusively on important concepts, we also filtered out short one-word entities (e.g.: *cell, administration*) (we manually experimented with different thresholds for filtering out one-word entities and got the best results by filtering out ones that are shorter than 14 characters). Overall, 8'782 different entities and 11'797 different entity labels were extracted. After this preprocessing step, 38'025 distinct relations among entities were created.

4.4.3 ScienceWISE Collection

The second collection we consider is a testset for WSD we created based on public data obtained from the ScienceWISE system. The ScienceWISE system allows a community of scientists, working in a specific domain, to generate dynamically as part of their daily work a field-specific ontology with direct connections to research papers and scientific data management services. The main functionalities of ScienceWISE are *discovery of relevant scientific papers*, *annotations* (i.e., adding meta-data to scientific documents) and *semantic bookmarking* (i.e., creating virtual collections of research papers from arXiv).

The domain-specific ontology is central to the system and allows to integrate all heterogeneous pieces of data and content shared by the users. Since the underlying scientific domain of the ontology is often rapidly changing and only loosely-defined, the best way to keep it up-to-date is to crowdsource its construction through a community of expert scientists. The initial version of the ontology was created by performing a semi-automated import from many science-oriented ontologies and online encyclopedias. After this step, ScienceWISE users (who are domain experts) were allowed to edit elements of the ontology (e.g., adding new definitions or new relations) in order to improve both its quality and coverage. Presently, the ScienceWISE ontology, which is publicly available in RDF[6], counts more than 60'000 unique entries, each with its own definitions, alternative labels, and semantic relations.

Using documents and human-created annotations over the ScienceWISE ontology, we created a testset for WSD. The generated test collection contains 1) a set of 4'691 abstracts from the Physics domain, 2) a set of 5'217 disambiguation decisions performed by experts in the Physics domain, and 3) the version of the ScienceWISE ontology as of October 2012 which has been used for our domain-specific ontology-based WSD approach.

Formally, a test collection TC is represented as the following set of tuples: $TC = \{(d, e_a, e_u) | d \in D, e_a \in E_A, e_u \in E_A\}$, where e_a and e_u represent the ambiguous and unambiguous (main)

[4] http://bioportal.bioontology.org/annotator,
REST API description: http://rest.bioontology.org/

[5] The exact version of the ontology we used is *2012_2011_09_09*.

[6] http://sciencewise.info/ontology/

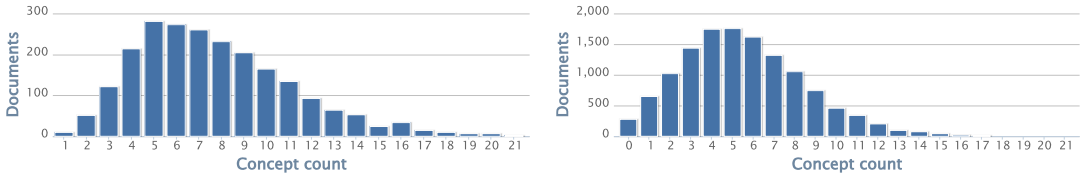

Figure 4.1 – Distribution of entities per document in the ScienceWISE (left) and MSH (right) collections.

labels of the same entity respectively. Both collections with detailed descriptions are available online[7] for reproducibility purposes.

4.4.4 Concept Extraction and Distribution

Given the plain text abstracts and the corresponding ontology, we build the DECVs for both collections as follows. First, we create an index from all the scientific concepts (entities) appearing in the collection ontology by considering stemming (Porter stemming algorithm) and stopword removal. Then, we process each abstract and match its textual contents to the entity index using an efficient and exact string matching method and using TF-IDF as scoring function. The final distributions of concepts for the MSH and the ScienceWISE documents are depicted in Figure 4.1. As we can observe, most abstracts contain 5-6 entities in the ScienceWISE collection and 4-5 entities in the MSH collection.

4.4.5 Experimental Results

In scientific articles, acronyms are often used to shorten commonly used concepts across the document. Usually, such acronyms are defined the first time they appear in the paper (e.g., Color Dipole Model (CMD)). Those occurrences make it easy to automatically detect the right sense of such ambiguous acronyms by simply using regular expressions to look for the definition given before or after the brackets. Using simple regular expressions, we discovered that we can directly solve 56% of the cases in the ScienceWISE collection and 67% in the MSH collection. Thus, we divide our test collection TC into 2 sub-collections $TC_R \cup TC_U = TC$ that represent the sub-collection containing the cases that can be simply resolved and the other cases respectively. For this reason, we report in the following the effectiveness of the proposed methods on the sub-collection TC_U. The supervised NB method is trained over the sub-collection TC_R.

Table 4.3 gives the effectiveness values for ECV approaches as compared to our baselines for the two test collections. Among the baselines, we observe that the supervised NB performs best on ScienceWISE.

[7] https://github.com/XI-lab/entity-disambiguation-data-ecir2013

Table 4.3 – Precision of context vector based disambiguation over the subset of concepts that cannot be disambiguated using regular expressions. We indicate statistical significant improvements over NB with * and over the best unsupervised baseline with $^+$.

WSD Approach	Precision (ScienceWISE)	Precision (MSH)
Random	39.97	46.73
Most Frequent	74.46	43.60
Context Vectors	74.29	**95.29**
NB	**85.13**	67.31
TF-IDF ECV	80.72^+	90.46^*
Binary ECV	$\mathbf{93.34}^{*+}$	$\mathbf{90.77}^*$

Table 4.4 – Precision of ontology graph based WSD.

WSD Approach	Precision (ScienceWISE)	Precision (MSH)
minDist (with Cat)	**88.82**	-
OSP (with Cat)	86.46	-
NN (with Cat)	73.93	-
minDist (without Cat)	**82.84**	67.28
OSP (without Cat)	77.42	56.77
NN (without Cat)	73.93	**72.37**

Analyzing the results of the proposed ontology-based approaches, we see that ECV-based approach outperforms basic unsupervised approaches and is comparable to supervised approaches (i.e, NB). Specifically, we note that ECVs outperform standard Context Vectors in terms of effectiveness while also being more efficient in terms of indexing as the term space is considerably reduced since it only considers entities in the ontology instead of all terms in the document collection.

On the MSH collection, Context Vectors perform best overall. This can be explained by the relatively low quality of its background ontology which has been automatically constructed. On the other hand, in ScienceWISE the ontology is manually built and curated by a community of domain experts which makes the approaches exploiting such information perform best. This hypothesis is supported when we look at the Precision/Coverage graph (Figure 4.2), where we observe that by lowering the coverage of matching entities, Precision of ECVs becomes greater than Precision of CV also for the MSH collection.

Moreover, we note that for the ScienceWISE collection, the simpler Binary ECV approach (which considers only binary values in the vectors indicating co-occurrences) performs better than the TF-IDF ECV method, which instead uses TF-IDF scores for the concept context vectors. TF-IDF ECVs perform best however on the MSH collection, albeit by a small margin (less than 0.3%).

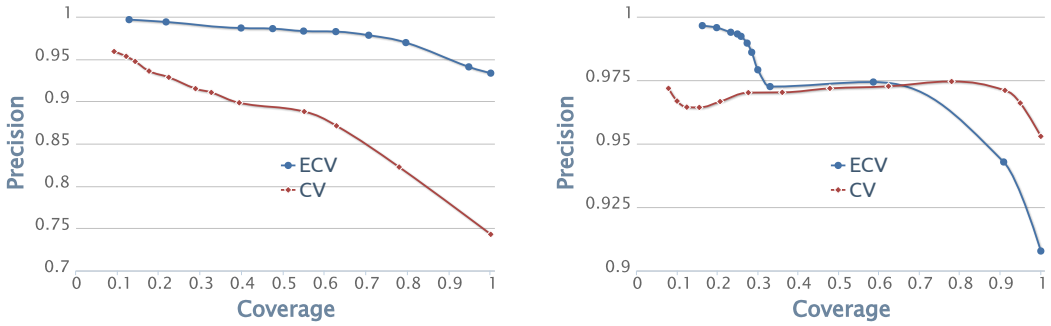

Figure 4.2 – Precision/Coverage graphs for CV and Binary ECV methods over the ScienceWISE (left) and MSH (right) collections.

Table 4.5 – Precision of combined WSD semantic approaches. We indicate statistically significant improvement over Binary CCV with *.

WSD Approach	Precision (ScienceWISE)	Precision (MSH)
Binary ECV	93.34	**90.77**
+ minDist (with Cat)	92.68	77.56
+ OSP (with Cat)	94.44	80.77
+ NN (with Cat)	**94.53***	90.60

Table 4.4 presents the results for WSD approaches based on the ontology graph. While they also outperform unsupervised and, in some cases, supervised baselines, they are not better than ECV-based approaches. Among the graph-based approaches, the NN method yields the best effectiveness over the MSH collection while NN performs best on ScienceWISE. For the ScienceWISE dataset, we run our approaches over two different versions of the ontology graph: one that includes the edges about category information (similarly to Wikipedia articles and categories) and one containing exclusively edges that relate entities to each other (note that for MSH the category information is not available). We observe that considering category links provides better WSD effectiveness. Thus, we only report results using the more complete ontology graph for ScienceWISE in the following.

Next, we evaluate the combination of ontology-based approaches to WSD. Specifically, Table 4.5 shows the combination of Binary ECV with methods based on the ontology graph. The methods are combined using the model from Equation 4.4 using equal weights for all the components. As we can see, the combination of ECVs and graph-based NN method outperforms both individual approaches on the ScienceWISE collection. On the MSH collection, no significant improvement is observed.

Table 4.6 – Execution time of different WSD approaches over the two test collections.

WSD Approach	Exec Time (ms) (ScienceWISE)	Exec Time (ms) (MSH)
Context vectors	14'712 (+30 min indexing)	1'826 (+1 h indexing)
ECV	1'682 (+2 min indexing)	1'476 (+5 min indexing)
NN	35'363	41'947

Parameter Sensitivity in the Mixture Model.

As described in Section 4.3.4, we combine different approaches by considering a linear combination of their confidence scores. Figure 4.3 gives the results of a parameter sensitivity analysis we performed for such combinations. The figure shows precision values for the combination of the Binary ECV method with two different approaches exploiting the ontology graph, namely, NN and OSP. As we can see, optimal effectiveness values are obtained when more weight is put on matching evidence coming from the graph. Considering an equal weight is hence somewhat suboptimal, though it also results in high effectiveness values.

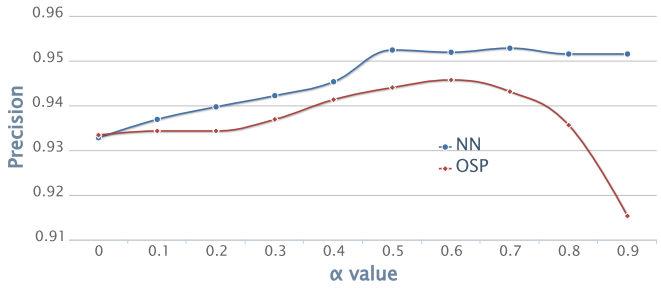

Figure 4.3 – Precision values varying the α parameter of the mixture model in Equation 4.4.

Efficiency Considerations.

As very large collections of scientific documents are available in digital libraries, in addition to WSD effectiveness we are also interested in how efficient our methods are when deployed in large-scale, real settings. Table 4.6 reports the execution times of different WSD methods over the two test collections.

We observe that the running times of graph-based approaches are higher than those relying only on vector similarities, mainly because of the costly access times to the database system. However, the preparation of both term and entity vectors requires considerable time, which is not needed by the graph-based approaches.

4.5 Conclusions

Scientists originating from different sub-communities often use the same term to refer to different concepts, making it hard to automatically process their articles using simple NLP or indexing techniques. In this chapter, we tackled the problem of correctly disambiguating terms appearing in the abstracts of scientific publications using a series of techniques ranging from relatively simple approaches (e.g., most common sense) to several variants of context vectors and to a series of new ontology-based approaches we devised for this work.

While creating and maintaining a field-specific ontology represents a huge effort, more and more scientific portals rely on such ontologies to organize their contents (the two ontologies we used in the context of this chapter are good examples of that trend). Following our experiments, we observe that such ontologies can represent crucial information when building word sense disambiguation systems, for two main reasons: i) ontologies typically regroup the most important terms of a scientific domain and can thus be used to build more efficient and effective context vectors based on ontological entities only and ii) the structure of the ontology can be leveraged to devise new techniques for WSD, for example using distance measures or nearest-neighbors on the ontology graph. Combining entity context vectors and graph-based approaches yields the best results according to our experiments, where our combined methods outperform both Bayes classifiers and conventional context vectors when leveraging on a high-quality and relatively complete ontology.

In the next chapter, we address another complimentary task for knowledge extraction and ontology maintenance, namely coreference resolution. Coreference resolution allows to identify new unseen labels for entities within documents, as well as other entity mentions that are not labels, such as pronouns. We apply a novel semantic coherence pipeline on top of an existing state-of-the-art coreference resolution system to improve its performance.

5 Ontology-based coreference resolution

5.1 Introduction

Natural language understanding is often referred to as an *AI-complete* task, meaning that it belongs to the class of the most difficult problems in Artificial Intelligence, which would require machines to become as intelligent as people prior to being solved. While perfect natural language understanding is still out of reach, recent advances in machine learning, entity linking, and relationship mining are closing the gap between humans and machines when it comes to processing natural language. Semantic technologies have played a key role in those developments, by providing mechanisms to classify, describe, and interrelate entities using machine-processable languages.

Less attention has however been given to the problem of leveraging Semantic Web techniques and knowledge bases to find all expressions referring to the same entity in a text, i.e., *coreference resolution*. While a flurry of previous contributions have proposed techniques to resolve coreferences (see Section 5.2 below), the extent to which semantic technologies can be leveraged in this context remains unclear. In this chapter, we investigate this question and introduce `SANAPHOR`, a new system focusing on the last stage of a typical coreference resolution pipeline and improving the quality of the coreference clusters by exploiting semantic entities and fine-grained types to split or merge coreference clusters.

The following piece of text, for example, motivates our approach:
"Laiwu City of Shandong Province has established a cell structure cultivation center ... currently Shangong has established ten agricultural development and model zones similar to that of Laiwu City."
With purely syntactic and grammatical approaches, it is easy to get confused between the name of the province and the name of the city, since they initially appear together. In fact, `Stanford Coref` will put occurrences of both the province and the city into one coreference cluster. Access to external knowledge such as ontologies or knowledge bases is key in this context.

In the following, we add a semantic layer on top of the prominent `Stanford Coref` pipeline [114][1] to tackle such cases. Throughout our process, we leverage a number of state-of-the-art Semantic Web techniques ranging from entity linking to type ranking. We focus on type-based coreferences, excluding part-of-speeches that do not possess self-contained semantics (e.g. determiners, pronouns etc).

In summary, the contributions of this work are:

- A new system that adds a semantic layer to the state-of-the-art `Stanford Coref` pipeline.
- A novel NLP technique that leverages the semantic web to better resolve coreferences.
- An empirical evaluation of our system on standard datasets showing that our techniques consistently improve on the state-of-the-art approach by tackling those cases where semantic annotations can be beneficial.

The rest of this chapter is structured as follows: in the rest of this section we define the concepts of coreference and anaphora by presenting several examples; in Section 5.2 we discuss related work in Semantic Web technologies and on coreference resolution systems; Section 5.3 describes the architecture of the system we propose; finally, Sections 5.4 and 5.5 describe the experimental evaluation of `SANAPHOR` and conclude the paper.

5.1.1 Preliminaries

We start below by introducing the terminology used throughout the rest of this chapter. Some of the linguistic units appearing in textual contents have the function of representing physical or conceptual objects. Linguists often call such units *referring expressions*, while the objects are called *referents* and the relations that unite a referring expression and its referent are called *references*. In the following example: *So Jesus said again, "I assure you, I am the gate for the sheep. All those who came before me were thieves and robbers. [...] I have other sheep too. They are not in this flock here."* the referring expressions are:

- Noun Phrases (NPs) and pronouns referring to people (e.g. *Jesus* ; *all those who came before me*), things (*the gate*), classes (*sheep*; *they*) or that designate interlocutors (*I*; *you*)
- clauses, that names facts (*I am the gate for the sheep*; *I have other sheep too*; *they are not in this flock here*)
- the adverb *here* that designates a location.

In order to satisfy cohesion [55], the same object is often recalled throughout the text repeatedly so that it can be enriched with new attributes.

[1] http://nlp.stanford.edu/projects/coref.shtml

In this context, linguists often distinguish *coreference* from *anaphora*. The difference between the two concepts is subtle and is explained in the following. We have a *coreference* every time two (possibly different) referring expressions denote *the same referent*, that is, the same entity. For example, in the sentence *George Washington, the first president of the USA, died in 1799.*, "George Washington" and "the first president of the USA" refer to the same entity, thus, they co-refer. We have an *anaphora* every time the reference of an expression $E2$, called *anaphoric expression*, is function of a previous expression $E1$, called *antecedent*, so that one needs $E1$ to interpret $E2$. For example, in the sentence *I like birds! Those animals are really cute!* "those animals" is an anaphoric expression and the reader needs to know that it refers to "birds" (the antecedent) in order to understand the sentence. Finally, the two concepts can be combined:

- The sentence *You have a cat? I like them.* is a case of anaphora without coreference since the pronoun *them* needs the antecedent *a cat* to be interpreted (it is the anaphoric), but the two references do not designate the same object (*a cat* = an individual / *them* = the entire species).
- The sentence about George Washington we presented before is an example of coreference without anaphora, since if we remove "George Washington" one can still understand the sentence.
- The sentence *The dragon is coming. It is going to burn the city!* is an example of anaphora and coreference since one needs an antecedent to resolve "It", and both "It" and "the dragon" refer to the same entity.

In this work we show how entity types can be used in order to resolve the two last cases.

5.2 Related Work

5.2.1 Named Entity Recognition

Named entity recognition (NER) refers to the task of correctly identifying words or phrases in textual documents that represent entities such as people, organizations, locations, etc. During the last decades, NER has been widely studied and the best NER approaches nowadays produce near-human recognition accuracy for generic domains such as news articles. Several prominent NER systems employ supervised learning methods based on maximum entropy [19] and conditional random fields [40], or fuse the results of other systems using a supervised classifier [123].

5.2.2 Entity Linking

Entity linking is the task of associating a textual mention of an entity to its corresponding entry in a knowledge base. It can be divided into three subtasks: mention detection, link generation,

and disambiguation [87]. One of the main issues that needs to be tackled when doing entity linking is the ambiguity of the textual representation of the entity given as input. For example, the mention "Michael Jordan" can be linked to both Michael Jordan the basketball player and Michael Jordan the well-known machine learning professor. Much work has been done on entity linking. Recently, Houlsby and Ciaramita [63] dealt with ambiguities by using a variant of LDA in which each topic is a Wikipedia article (that is, an entity). Cheng and Roth [24] used Integer Linear Programming to combine relational analysis of entities in the text, features extracted from external sources and statistics on the text.

In the context of this chapter, both NER and Entity Linking are prerequisites for coreference resolution as we take advantage of external knowledge to improve the resolution of coreferences and hence must first identify and link as many entity mentions as possible to their counterparts in the knowledge base. Since, however, those two tasks are not the focus of this work, we decided to use the TRank [138] pipeline because of its simplicity and its good performance in practice on our dataset (see Section 5.4).

5.2.3 Entity Types

Knowing the types of a certain entity is valuable information that can be used in a variety of tasks. Much work has been done on extracting entity types both from text and from semi-structured data. In this context, Gangemi *et al.* [44] exploit the textual description of Wikipedia entities to extract entity types, Nakashole *et al.* [98] designed a probabilistic model to extract the types out of knowledge base entities, and Paulheim and Bizer [104] worked on adding missing type statements by exploiting statistical distributions of types as subjects and objects of properties. Much effort has been put also on ranking entity types in several contexts. TRank [138] is a system for ranking entity types given the textual context in which they appear. Tylenda *et al.* [143] select the most relevant types to summarize entities. In this work we leverage entity types as evidences for deciding if, given a piece of text, different entity mentions refer to the same entity or not.

5.2.4 Coreference and Anaphora

According to Ng [101], practically all coreference and anaphora resolution systems are instantiations of a seven-step generic algorithm[2]:

1. Identification of referring expressions: This first step is mostly to identify all of the pronouns and noun phrases in the text. Clauses and adverbs can also be spotted.
2. Characterization of referring expressions: This second step consists of determining and computing the information regarding referring expressions that might be relevant to its linking to another expression in the text. Most approaches rely on some preprocessing modules (e.g. part-of-speech tagging, parsing, named entity recognizer, etc.) to perform this step;

[2]Note that steps 3, 5 and 6 can be absent in a coreference or anaphora resolution algorithm. Moreover, existing algorithms differ in the way these seven steps are implemented

however, they differ in the level of sophistication of the extracted information, ranging from knowledge-rich to knowledge-poor (see below).
3. Anaphoricity determination: Involves distinguishing anaphoric expressions, that should have an antecedent, from non-anaphoric expressions, that should not. Thus, this step is always performed as part of anaphora resolution, but not always for coreference resolution (Section 5.1.1).
4. Generation of antecedent candidates: This fourth step identifies a set of potential antecedents, named *candidates*, that linearly precedes the anaphoric expression in the text.
5. Filtering: This step involves removing from the set some unlikely candidates based on ensemble of hard constraints, for example morphologic, syntactic and semantic constraints.
6. Scoring/Ranking: The aim of this step, that is optional, is to rank remaining candidates according to an ensemble of soft constraints, also called *preferences*, that often depend on psycholinguistic and discourse principles (especially *focus* [132], *centering* [49] or *accessibility* [8]).
7. Searching/Clustering: Finally, the goal of this last step is to select an antecedent for a given anaphoric expression from the set of candidates returned by the fifth and/or the sixth steps. If step 6 has been performed, then *searching* becomes the task of selecting the highest-ranking element in the candidate list; otherwise, the "best" expression is selected as the antecedent in accordance with criteria specified by the resolution algorithm. In the case of coreference resolution, this process corresponds to applying a single-link clustering algorithm to each anaphoric expression to cluster the referring expressions in the document and generate a partition.

Although this generic algorithm characterizes most of the resolution pipelines, research on coreference and anaphora resolution has been proceeding in many different directions for the last 30 years. Nevertheless, it is possible to identify important trends [101, 34, 103]. In the context of this work, two trends are of particular significance and are presented below.

First, coreference and anaphora resolution systems can be classified with respect to the types of knowledge sources they leverage. One typically differentiates *Knowledge-rich* systems from *knowledge-lean* systems. Early anaphora resolution systems [48, 133] as well as more recent ones [135, 109, 102, 52, 21, 145] are knowledge-rich systems that rely on domain informations (such as FrameNet, WordNet, Wikipedia, Yago, etc.), semantic and discourse analysis, and sophisticated inference mechanisms (induction for example). Knowledge-lean systems instead rely only on morphological and possibly syntactic information [78, 12, 94, 114], and reach high performance without semantic and world knowledge. Our system belongs to the first category, using YAGO and DBpedia.

Early coreference and anaphora resolution systems also differ from more recent ones by the fact that they adopt *knowledge-based approaches*, in which the rulesets used in *filtering* and *scoring/ranking* (see steps 5 and 6 above) are based on a set of hand-coded heuristics that specify whether two referring expressions can or cannot have any coreferential/anaphoric relationship [59, 49]. Actually, these approaches are often called *linguistic approaches* as they

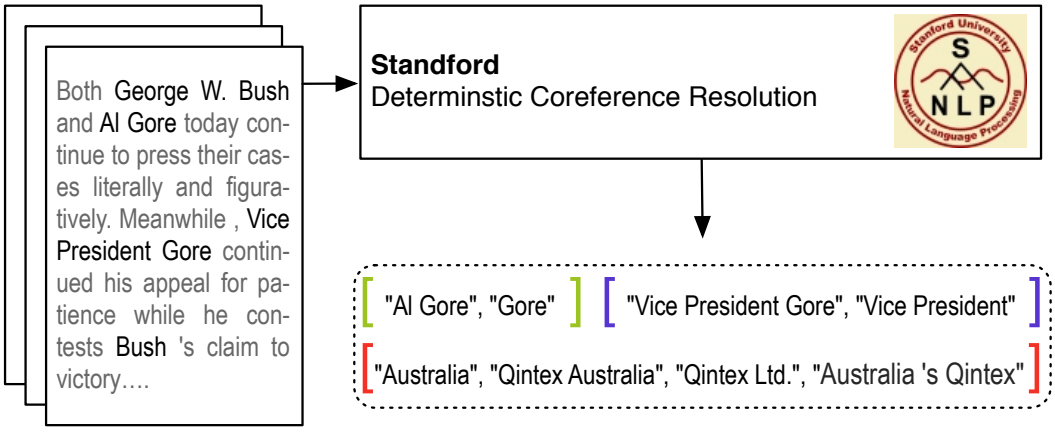

Figure 5.1 – The `Stanford Coref` system takes plain text as an input and outputs clusters (`[]`) of mentions (`""`) which are potentially coreferenced.

are based on linguistic theories. In contrast, *corpus-based approaches* acquire knowledge using a learning algorithm and training data, i.e., a corpus annotated with coreference and anaphora information in *filtering* and *scoring/ranking* [45, 57, 134]. Again, our own system belongs to the first category.

5.3 System Architecture

In this section, we describe the overall architecture of `SANAPHOR` and provide details on each of its components.

5.3.1 System Input

Starting from the `Stanford Coref` framework [114] (Figure 5.1), which covers the steps 1-7 described in Section 5.2.4, we obtain for each document (e.g., a news article) a set of clusters containing textual mentions. The clusters are non-overlapping and contain potentially co-referring mentions. In addition, `Stanford Coref` associates a headword with each mention (especially for long mentions) when possible.

5.3.2 System Overview

Many potential improvements are conceivable throughout the generic pipeline introduced in Section 5.2.4. In that context, our efforts first focused on improving coreference resolution using semantic word and phrase similarities based on Word Vectors [90]. However, word vectors did not work well in our experiments. For example, the vector of the word “shepherd” was very close to the vector of “sheep”, which is reasonable, but does not work well for the

coreference resolution task, since these two words often appear in one document. Motivated by the results analysis presented above, SANAPHOR focuses instead on splitting and merging of candidate clusters (see Step 7 in Section 5.2.4) using semantic information, as it is (in our opinion) the most susceptible to benefit from a tight integration of semantic technologies.

Figures 5.2 and 5.3 give an overview of our system, illustrating the preprocessing steps and the splitting/merging steps respectively. SANAPHOR receives as input the clusters of coreferences generated by `Stanford Coref`. Each cluster is a set of mentions extracted from the original text. Each mention comes in the form of a string and, potentially, an associated headword (the most salient word in the mention). The mentions can be either Named Entities, pronouns, or determiners, as identified and clustered by `Stanford Coref`. Our system then takes those clusters and proceeds in two successive steps I) Preprocessing, where we leverage linked data to represent named entities with their semantic counterparts (either Entities or Types) whenever possible; II) Cluster Optimization, where using annotations obtained from the preprocessing step we derive a strategy for splitting clusters containing unrelated mentions,

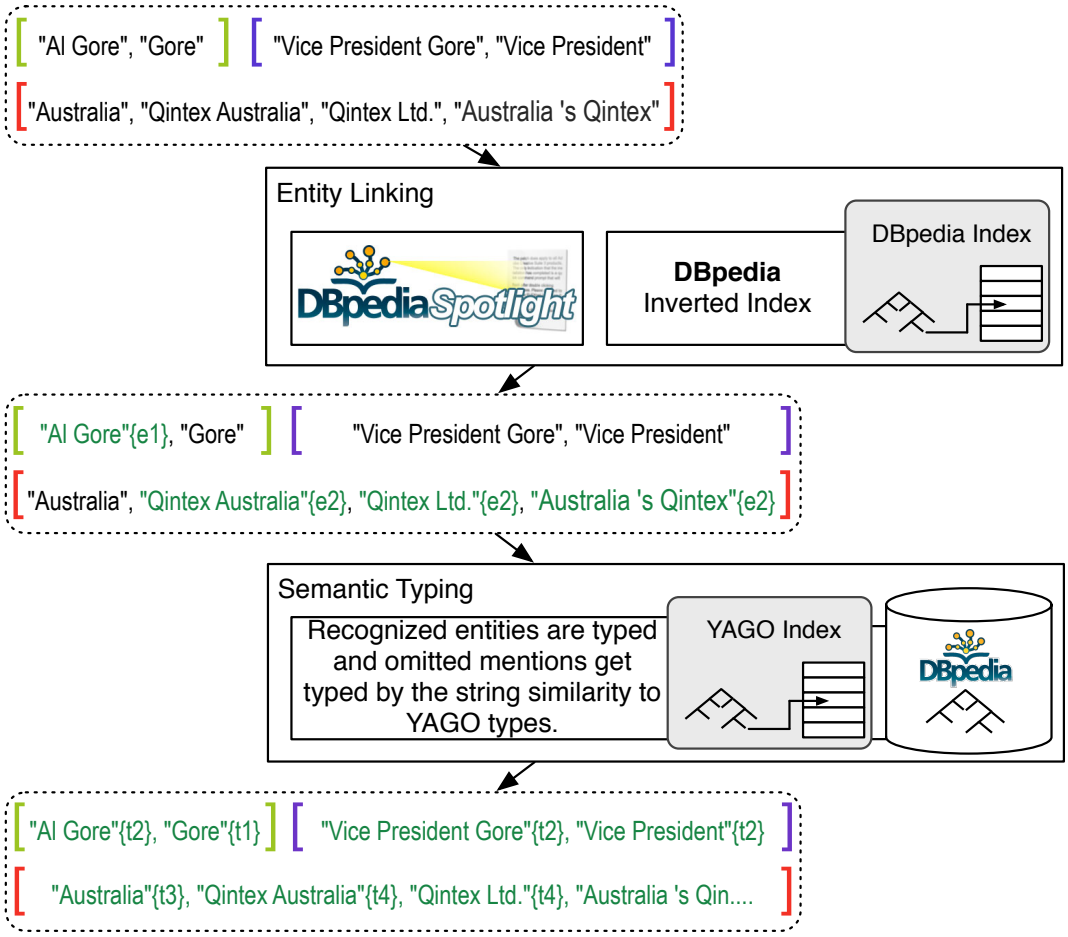

Figure 5.2 – The pre-processing steps of SANAPHOR that add semantics to the mentions.

or, conversely for merging mentions that semantically should belong together.

We describe in more detail the functionalities provided by those components in the following, starting with the semantic annotation pipeline and then moving to cluster management methods.

5.3.3 Semantic Annotation

Entity Linking

The goal of the Entity Linking component is to link entity mentions to DBpedia entries. We exploit an inverted index associating DBpedia labels to entity URIs. In order to generate

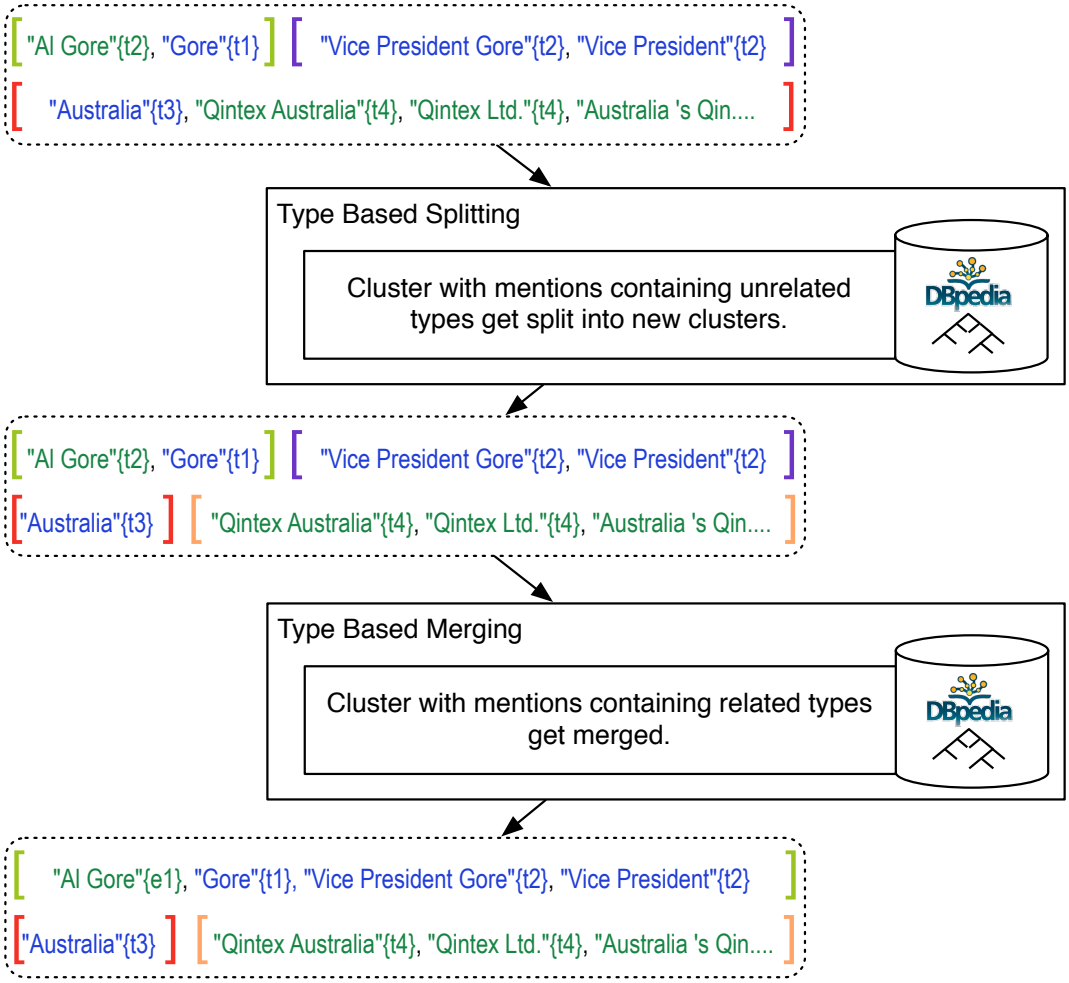

Figure 5.3 – The final type-based splitting and merging of the clusters in SANAPHOR.

high-quality links, we decided to only link mentions that exactly match DBpedia labels[3]. Entities with multiple aliases are handled using Wikipedia redirect links and, in order to foster precision, by discarding URIs that link to ambiguous entities (i.e., entities having a `wikiPageDisambiguates` property).

Semantic Typing

The next step in our preprocessing pipeline is assigning *Types* to mentions appearing in the text. In this context, we use the YAGO ontology[4] as a target knowledge base, as it is one of the largest type ontologies by the number of types. We created an inverted index of the types obtained from the YAGO ontology and performed a string matching between every mention and the inverted index. For example, a noun phrase "rock singer" is typed as ⟨Wikicategory_American_Rock_Singers⟩. For the mentions linked in the previous step, we employ the mappings between DBPedia and YAGO ontologies provided by TRank Hierarchy [138] to map DBPedia types to YAGO ones.

We chose to optimize our preprocessing steps for precision rather then recall, since the subsequent steps rely on precise linking to be effective at improving the mention clusters. As a result, we do not annotate labels that refer to multiple entity types.

5.3.4 Cluster Management

Splitting Coreference Clusters

The first task `SANAPHOR` undertakes to optimize the clusters of mentions is to split clusters containing mentions of different types. This step tackles cases where `Stanford Coref` was not able to deal with ambiguity in the text, for example for the following cases: "Aspen" (the city in Colorado) and "Aspen" (the tree), which can be wrongly interpreted as referring to the same referent, thus producing a series of incorrect coreferences. Instead, `SANAPHOR` leverages the output of the entity linking process to resolve the ambiguity of the mentions: since during the linking phase the two mentions will probably be associated to different entities, the system can decide to split them into separate clusters.

The result of the semantic annotation phase is a series of sets $\{\mathscr{S}_0, \ldots, \mathscr{S}_n\}$, one per coreference cluster, containing entities $e \in \mathscr{E}$ and/or fined-grained semantic types $t \in \mathscr{T}$ attached to each mention $m \in \mathscr{M}$. The splitting process examines all pairs of mentions $\{m_i, m_j\}$ in a given cluster, and decides whether or not to split the cluster depending on the potential entities $\{e_i, e_j\}$ and types $\{t_i, t_j\}$ attached to the mentions. Formally, we split a cluster whenever, $\forall \{m_i, m_j\} \in S$:

[3] We have also tried more complex methods that take context into account, such as *DBPedia Spotlight*, but they lead to less precise linkings and worse overall results.

[4] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/

- $\exists\{e_i, e_j\} \mid e_i \neq e_j$ or

- $\exists\{t_i, t_j\} \mid t_i \not\leq t_j$ (where $\not\leq$ stands for equivalence or subsumption relation w.r.t. the type hierarchy of the ontology), or

- $\exists\{e_i, t_j\} \mid T(e_i) \not\leq t_j$ (where $T(e_i)$ stands for the type of e_i according to the ontology).

Since a coreference cluster might also contain non-annotated mentions, we need a way to properly assign them to the split clusters. In order to do this, we first identify the words that belong exclusively to one of the mentions m_i or m_j. We assign all other mentions to one of the new clusters based on the overlap of their words with the exclusive words of each new cluster.

However, these steps alone do not systematically result in a substantial performance increase due to many possible reductions of the original mention. For example, a text might contain "Aspen Airways" first and then have the word "Aspen" to refer to the airline, which our method might incorrectly link to a city or a tree type. To overcome this problem, we introduce a simple heuristic that ignores entity linkings of the mentions whose words represent a complete subset of any other mention in the same cluster.

Merging Coreference Clusters

The second task that we are tackling in the context of cluster management is merging, that is, joining pairs of sets $\{S_i, S_j\}$ that contain similar entities or types. For instance, consider the mention "Hosni Mubarak", the former president of Egypt, which can also be referred to as "President Mubarak" in a news article. In such a case, `Stanford Coref` might assign those two mentions to two different clusters. Thus, starting from entity and type linking as before, we propose to merge clusters, each of which contains at least one mention that refers to the same entity. Formally, two sets $\{S_i, S_j\}$ corresponding to two clusters are merged whenever:

- $\exists\ (e_i \in S_i \wedge e_j \in S_j) \mid e_i \equiv e_j$ or

- $\exists\ (e_i \in S_i \wedge t_j \in S_j) \mid T(e_i) \leq t_j$ and when the condition just above does not apply.

We note that in this step we do not use any heuristic to pre-filter the clusters.

Our system, `SANAPHOR`, is available as an open-source[5] extension to `Stanford Coref`. The pipeline allows to use different entity and type linkers for future experiments.

[5] http:///github.com/xi-lab/sanaphor

5.4 Experimental Evaluation

5.4.1 Datasets

We evaluate our system on standard datasets from the CoNLL-2012 Shared Task on Coreference Resolution [111] distributed as a part of the OntoNotes 5 dataset[6]. We use the English part of the dataset which consists of over one million words from newswire, magazine articles, broadcast news, broadcast conversations, web data, telephone conversations and English translation of the New Testament.

The English dataset is split into three sub-collections: development, training and test. The development dataset is intended to be analyzed during the development of the coreference resolution system in order to build intuitions and tune the system. The training dataset is designed to be used in the supervised training phase, while the final results have to be reported on the test dataset. In the following sections, we analyze results and we design our methods based on the development collection and report the final results based on the test collection. Since our system improves on the Stanford Coreference Resolution System, which already includes supervised models, we do not directly use the training sub-collection in our pipeline.

5.4.2 Metrics

Many metrics have been proposed to evaluate the performance of coreference resolution systems, from early metrics like MUC [147], to the most recent metric proposed—BLANC [122].

As a final evaluation metric, we use the most recently proposed BLANC, which addresses the drawbacks of previously proposed metrics such as MUC, B-cubed [11], or CEAF [84], as it neither ignores singleton mentions nor does it inflate the final score in their presence.

In addition, we employ a standard pairwise metric based on the *Rand Index* [117] to evaluate the performance of the cluster optimization steps of our system in isolation.

5.4.3 Analysis of the Results of Stanford Coreference Resolution System

We start by analyzing the results of the `Stanford Coref` on the development dataset in the context of two possible error classes: 1) mentions that were put into one cluster, but that in fact belong to different clusters, 2) mentions that refer to the same thing, but that were put into different clusters. Additionally, since we focus on noun-phrase mentions, we want to see how many noun-only clusters exist in the dataset in order to estimate the effect of a possible improvement.

Overall, the `Stanford Coref` system created 5078 coreference clusters, out of which 270 clusters need to be merged and 77 “has-to-be-merged” clusters are noun-only. The total

[6] https://catalog.ldc.upenn.edu/LDC2013T19

	0 Links	1 Distinct Link	2 Distinct Links	3 Distinct Links
All Clusters	4175	849	49	5
Noun-Only Clusters	1208	502	33	2

Table 5.1 – Cluster linking distributions for all the clusters and for noun-only clusters

number of clusters that should be split is 118, out of which 52 are noun-only.

As we can observe, the total amount of potential split and merge clusters account for approximately 8% of total data, which can result in a significant performance improvement for coreference resolution (for which even small improvements are considered as important given the maturity of the tools developed over more than 30 years).

In the following, we report results for the different steps in our pipeline on the test dataset.

5.4.4 Preprocessing Results

The main innovation of `SANAPHOR` is the semantic layer that enhances classic coreference clustering, hence we focus on evaluating clusters that contain at least one entity (or one type) at the output of our preprocessing steps. The overall recall of our approach is therefore bound by the number of clusters that were identified as containing linked entities and/or types.

In total, we linked 2607 mentions out of 9664 noun phrase mentions (i.e., mentions that have nouns as headwords) extracted by `Stanford Coref` from the **CoNLL dev dataset**. Out of these 9664 mentions, 4384 were recognized by `Stanford Coref` as entities. Table 5.1 summarizes the distribution of clusters and the links obtained using our preprocessing step.

For evaluation purposes, we consider only clusters that contain at least one link. Moreover, we make the following distinction of clusters for evaluation purposes:

- **All Linked Clusters.** That is, clusters that contain at least one linked mention, or
- **Noun-Only Linked Clusters.** These are clusters which contain at least one linked mention, headword, but have no pronouns nor determiners.

We make this distinction in order to evaluate whether considering clusters with pronouns and determiners (which bear little semantic information) affects the overall results.

5.4.5 Cluster Optimization Results

Now, we turn our attention to the evaluation of the effectiveness of our cluster optimization methods (splitting and merging). The following experiments are performed on the CoNLL test dataset. We compute Precision, Recall and F1 metrics for the clusters on which we

		SANAPHOR			Stanford Coref		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Split	All Clusters	82.56	90.27	**86.25**	71.39	100.00	83.31
	Noun-Only Clusters	78.99	90.38	**84.30**	58.43	100.00	73.76
Merge	All Clusters	94.58	100.00	**97.21**	96.65	55.10	70.18
	Noun-Only Clusters	76.92	100.00	**86.96**	85.00	56.67	68.00

Table 5.2 – Results of the evaluation of the cluster optimization step (split and merge).

operate. Since we are evaluating clusters, we use the pairwise definition of the metrics (see Section 5.4.2).

We distinguish the results for both the split and merge operations as compared to the ground-truth. For instance, for all the clusters generated by each system, we perform pairwise comparisons of all mentions in the clusters and evaluate whether the two mentions were correctly separated (in case of a split) or put together (in case of a merge).

Table 5.2 summarizes the results of our evaluation. As can be seen, SANAPHOR outperforms Stanford Coref in both the split and merge tasks for both All and Noun-Only clusters. Moreover, we notice that the absolute increase in F1 score for the split task is greater for the Noun-Only case (+10.54% vs +2.94%). This results from the fact that All Clusters also contain non-noun mentions, such as pronouns, which we do not directly tackle in this work but have to be assigned to one of the splits nevertheless. Our approach in that context is to keep the non-noun mentions with the first noun-mention in the cluster, which seems to be suboptimal for this case.

For the merge task, the difference between All and Noun-Only clusters is much smaller (+27.03% for the All Clusters vs +18.96% for the Noun-Only case). In this case, non-noun words do not have any effect, since we merge clusters and also include all other mentions.

5.4.6 End-to-End Performance

Finally, and in addition to the previous results that reflect the effectiveness of SANAPHOR on relevant clusters, we evaluate the impact of our approach on the end-to-end coreference resolution pipeline using the CoNLL test collection. In that context, we use the Precision, Recall and F1 scores of the BLANC metric (Section 5.4.3). Our system consistently outperforms the Stanford Coref baseline in both Precision (60.63% vs 60.61%), Recall (55.16% vs 55.07%) and F1 values (57.11% vs 57.04%). The reason behind the limited improvement on the overall dataset is imputable to the recall we achieve during the linking step (see Section 5.4.4) and to the limited number of cases in which a split or a merge is required (8% of the total data).

To further elaborate on the significance of our results, we also ran our SANAPHOR pipeline

on the data where we annotated all entities with the "gold" (i.e., ground-truth) URLs. This corresponds to the optimal case where the system is able to link all possible entities correctly. The performance of `Stanford Coref` for such a best-case scenario is 57.17% in terms of F1, which is comparable to the performance of our entity linking method, thus confirming the validity of our approach.

5.5 Conclusions

In this chapter, we tackled the problem of coreference resolution by leveraging semantic information contained in large-scale knowledge bases. Our open-source system, SANAPHOR, focuses on the last stage of a typical coreference resolution pipeline (*searching and clustering*) and improves the quality of the coreference clusters by exploiting semantic entities and fine-grained types to split or merge the clusters. Our empirical evaluation on a standard dataset showed that our techniques consistently improve on the state-of-the-art approach by tackling those cases where semantic annotations can be beneficial.

Our approach can be extended in a number of ways. One of the limitations of SANAPHOR affecting its recall is due to the incomplete information in the knowledge base. In that sense, techniques that take advantage of a series of knowledge bases (e.g., based on federated queries), that identify missing entities in the knowledge base or that dynamically enrich the knowledge base could be developed. Another interesting extension would be to bring more structure to the coreference clusters, for example by introducing semantic links between the candidates in order to foster more elaborate post-processing at the merging step.

In the next chapter, we look at a higher-level task to annotate documents with tags, which are represented by named entities. We use data from a real-world system to propose better tag recommendation strategies and compare them with the state-of-the-art ones.

6 Applications in knowledge discovery for Scientific Literature

6.1 Introduction

The nature of scientific research is drastically changing. Fewer and fewer scientific advances are carried out by small groups working in their laboratories in isolation. In today's data-driven sciences (be it biology, physics, complex systems or economics), the progress is increasingly achieved by scientists having heterogeneous expertise, working in parallel, and having a very contextualized, local view on their problems and results. We expect that this will result in a fundamental phase transition in how scientific results are obtained, represented, used, communicated and attributed. Different to the classical view of how science is performed, important discoveries will in the future not only be the result of exceptional individual efforts and talents, but alternatively an emergent property of a complex community-based socio-technical system. This has fundamental implications on how we perceive the role of technical systems and in particular information processing infrastructures for scientific work: they are no longer a subordinate instrument that facilitates daily work of highly gifted individuals, but become an essential tool and enabler for performing scientific progress, and eventually might be the instrument within which scientific discoveries are made, represented and brought to use.

Any such tool should in our opinion possess two central components. One is a *field-specific ontology*, i.e., a structured organization of the knowledge created by the researchers in a given field, along with a formal description of the information and processes they utilize. While in some important cases (e.g., in bioinformatics or chemistry) it is possible to create large ontologies of sufficiently homogeneous concepts and automatically manipulate them using formal rules (see e.g. [125]), the ontology of scientific knowledge *per se* is very complex and vaguely defined at any given point in time. Scientific ontologies can therefore only be created by a combination of existing automatic methods and novel approaches that will enable human-machine collaboration between scientists and the knowledge management infrastructures allowing to combine presentation of new results, in-depth discussions, "user-friendly" introductions for young scientists, and meta-data to relate semantically similar

concepts or pieces of content. Today, there are no standard tools to insert, store and query such meta-data online, which mostly remains "in the heads of the experts" [2].

The organization of scientific information does not end with the generation of the scientific ontology. The second crucial element is a set of meaningful connections between such an ontology and the body of research material (papers, books, datasets, etc.). The challenge here is to connect semi-structured data to the natural language content of scientific papers through semantically meaningful relations. This creates a number of challenges to the current state-of-the-art in information retrieval, named entity recognition and disambiguation (since scientific concepts can have many different names and context-dependent meanings).

In this chapter, we tackle the problem of *ontology-based tagging*, i.e., of improving the relevance of automatically selected tags in large-scale ontology-based information systems. Contrary to traditional settings where tags can be chosen arbitrarily, we focus on the problem of recommending tags (e.g., scientific concepts) directly from a collaborative, user-driven ontology.

The contributions of this work are as follows:

- We formally define the task of ontology-based tagging and suggest standard metrics borrowed from Information Retrieval to evaluate it.
- We contribute a real document collection, a domain-specific ontology, and lists of expert-provided tags picked from the ontology and assigned to the documents as a standard evaluation collection for ontology-based tagging.
- We compare the effectiveness of standard Information Retrieval techniques (based on Term Frequency and Inverse Document Frequency) on our evaluation collection.
- We also compare the effectiveness of ontology-based techniques (e.g., based on ontological neighborhood or subsumption) and semantic clustering techniques (such as Latent Semantic Indexing and Dirichlet Allocation).
- Finally, based on the results of our experiments, we draw conclusions w.r.t. the practicality and usefulness of using a given technique for ontology-based tagging and discuss future optimizations that could be used to improve our results.

The rest of this chapter is structured as follows: We start by discussing related work in Section 6.2. We briefly present ScienceWISE, the infrastructure we leverage on for our experiments, and formally define the task we tackle in Section 6.3. We discuss our metrics and data sets in Section 6.4. We report on our experimental results and compare the effectiveness of various approaches for ontology-based tagging in Section 6.5, before concluding in Section 6.6.

6.2 Related Work

Research on tag recommendation can be classified into two main categories. A first class of approaches looks at the contents of the resources while a second one looks at the structure connecting users, resources, and tags. Examples of the former class include content-based filtering [93] and collaborative-filtering tag suggestion techniques [151]. Along similar lines, we previously experimented with tag propagation in document graphs in [22]. The latter class includes approaches that focus on the user rather than just providing tag recommendations given a resource. In [82], a set of candidate tags is created and then filtered based on choices made by the user in the past. FolkRank is an system based on a user-resource-tag graph [66]: it computes popularity scores for resources, users, and tags based on the well-known PageRank algorithm. The assumption is that the importance of resources and users propagates to tags.

Word sense disambiguation (WSD) is the task of identifying the correct meaning of an ambiguous word (e.g., "bank" can indicate either a financial institution or a river bank). A common technique for WSD is to exploit the context of ambiguous words, that is, other words in its vicinity (e.g., in the same sentence). An approach following this idea has been used by in [14], where among all the possible senses for a word in WordNet [38], the correct one is chosen by measuring the distance (based on text similarity functions) between the word context and its synsets (i.e., the set of all synonyms for one sense).

Though tag recommendation and disambiguation have been studied extensively (both for free-text tagging and folksonomy systems), surprisingly little research has been carried-out for tag recommendation and disambiguation in a Semantic Web context. Contag [4] is an early system recommending tags by extracting topics using online Web 2.0 services and matching them to an ontology using string similarity. To the best of our knowledge, the present effort is the first systematic and repeatable experimental study of tag recommendation for large-scale and collaborative ontology-based information systems.

6.3 The ScienceWISE system

The ScienceWISE system allows a community of scientists, working in a specific domain, to generate dynamically as part of their daily work an *interactive semantic environment*, i.e., a field-specific ontology with direct connections to research artifacts (e.g., research papers) and scientific data management services. The central use-cases of ScienceWISE are *discovery of relevant scientific papers*, *annotations* (e.g., adding "supplementary material" or meta-data to scientific artifacts) and *semantic bookmarking* (e.g., creating virtual collections of research papers from arXiv [3]).

The system has been public since 2010 and is accessible by scientists via the website[1], as well as via arXiv and the CERN Document Server[2]. The system currently counts several hundred

[1] http://sciencewise.info/
[2] http://cds.cern.ch

active users (using the services on a regular basis), thousands of annotated papers, and is continuously receiving several new registrations daily.

The domain-specific ontology is central to our system and allows us to integrate all heterogeneous pieces of data and content shared by the users. Since the underlying domain of the ontology is often rapidly changing and only loosely-defined, the best way to keep it up to date is to crowdsource its construction through the community of expert scientists. To create the initial version of the ontology, we have performed a semi-automated import from many science-oriented ontologies and online encyclopedias. After this initial step, ScienceWISE users (who are domain experts) are allowed to edit elements of the ontology (e.g., adding new definitions or new relations) in order to improve both its quality and coverage. Presently, the ScienceWISE ontology counts more than 60'000 unique entries, each with its own definitions, alternative forms, and semantic relations to other entries.

In the context of this work, we focus on two important and related problems that we have to tackle in order to improve the user experience: tag recommendation and tag disambiguation. We note that those two tasks are key not only in our setting, but for all large-scale, collaborative and ontology-based information systems that are currently gaining momentum on the Internet.

6.3.1 Tag Recommendation

When users bookmark an arXiv paper, our system attempts to automatically select the most relevant tags for characterizing the paper. The tags in question are in our case scientific concepts (entities) that are defined in the ontology. A user-friendly interface allows then to correct the system recommendation, e.g., by adding relevant tags or removing irrelevant tags from the top-k list that the system recommended.

More formally, the tag recommendation task can be defined as follows: a set of expert users bookmark scientific papers (documents) $\{d_1, \ldots, d_n\} \in D$. A ranked list of tags $(t_1^j, \ldots, t_{m_j}^j)$ is initially built for each paper d_j by selecting tags from entities in the ontology ($t_i^j \in \mathcal{T} \ \forall \ i, j$). This list is curated *a posteriori* by the expert users. We write T_{rel}^j to denote the set of relevant tags chosen by the experts for paper d_j. The other tags are defined as irrelevant: $T_{\overline{rel}}^j \equiv \mathcal{T} \setminus T_{rel}^j$.

6.3.2 Tag Disambiguation

The second problem we tackle is tag disambiguation. Since the same textual term can refer to several different entities, it is often difficult to disambiguate isolated terms appearing in a paper. For instance, if *anomaly* appears in the text of a scientific paper, should it be related to the *quantum anomaly* concept, to *experimental anomaly* or to *reactor neutrino anomaly*? All are valid scientific concepts but are however very different semantically. Similarly, depending on the context the abbreviation *DM* can mean *Dark matter* (cosmology), *Distance measure*

(astronomy), or *Density matrix* (statistical mechanics).

The goal of this second task is to detect such cases and to develop methods to effectively predict which entity an isolated mention should be associated with. Obviously, this second task directly relates to our first task, since disambiguating tags produces more relevant results and hence improves the quality of tag recommendation in the end. Formally, given a term τ appearing in the text of a paper and a set of automatically selected tags $\{t_1, \ldots, t_m\}$ corresponding to entities whose labels all contain the term τ, our goal is to automatically select the right tag(s) $t \in T^{\tau}_{rel}$ corresponding to the correct semantics of the term as chosen by our expert users.

6.4 Experimental Setting

6.4.1 Hypotheses

We consider the following hypotheses for the tag recommendation task: i) entities appearing in the title and the abstract of a paper are highly relevant to that paper, ii) excluding entities that are too generic yields better recommendations, and iii) using the structure of the ontology can help us recommend better tags. To evaluate those hypotheses, we compare eight different techniques in Section 6.5.1.

For the tag disambiguation task, we study whether applying clustering techniques on the papers using unambiguous entities as features allows us to disambiguate entities with a high accuracy. To evaluate this hypothesis, we test two clustering techniques (LDA and K-means) in Section 6.5.2.

6.4.2 Metrics

We evaluate the effectiveness of our approach using four standard metrics borrowed from Information Retrieval:

Precision@k defined as the ratio between the number of relevant tags taken from the top-k recommended tags for document d_j and the number k of tags considered: $P@k = \frac{\sum_{i=1}^{k} \mathbb{1}(t_i^j \in T_{rel}^j)}{k}$ (where $\mathbb{1}(cond)$ is an indicator function equal to 1 when $cond$ is true and 0 otherwise).

Recall@k defined as the ratio between the number of relevant tags in the top-k for document d_j and the total number of relevant tags: $R@k = \frac{\sum_{i=1}^{k} \mathbb{1}(t_i^j \in T_{rel}^j)}{|T_{rel}^j|}$

R-Precision defined as Precision@R, where R is the total number of relevant tags for document d_j: $RP = P@|T_{rel}^j|$.

Average Precision defined as the average of Precision@k values calculated at each rank where

a relevant tag is retrieved over the total number of relevant tags: $AP = \frac{\sum_{i=1}^{|T_{rel}^j|} P@i \ \mathbb{1}(t_i^j \in T_{rel}^j)}{|T_{rel}^j|}$.

Those definitions are valid for a single document only. In the following, we also report values averaged over the entire document collection, e.g., Mean Average Precision (MAP) defined as: $MAP = \frac{1}{n}\sum_{j=1}^{n} AP_j$. The metrics for tag disambiguation are derived similarly (see below Section 6.5.2).

6.4.3 Data Sets

We use real data as available on our platform for all our experiments. Our document collection contains all the articles bookmarked by our top-5 most prolific users. This represents 16'725 scientific papers and 15'083 tags representing 2'157 distinct scientific concepts (out of the 16'725 total number of concepts currently available in our field-specific ontology). If the same paper is bookmarked by more than one user, we take the *union of tags* as the relevant set of tags. For the tag disambiguation experiments, we based our experiments on 2'400 articles originating from 6 different top-categories on arXiv (400 articles per category). The experimental data we used for our experiments is available online[3] for reproducibility purposes.

6.5 Experimental Results

We report below on our techniques and experimental results for tag recommendation and tag disambiguation.

6.5.1 Recommending Tags

We compare eight different techniques for tag recommendation below. Most of our approaches are based on term-weighting [126], which is a key technique used in most large-scale information retrieval systems. Basic term-weighting works as follows in our ontology-based context. First, we create an index from the labels of all scientific concepts appearing in the ScienceWISE ontology by considering their stem using Porter's suffix stripping [110]. Then, for each new bookmarked paper, we analyze all the terms appearing in the paper. Given the importance of acronyms in scientific papers, we first determine whether the term is an acronym or not by inspecting its length, capitalization, and by trying to match it to known terms[4]. Two cases can occur at this point: i) if the term is an acronym, we consider it *as is* and try to match it to our concept index ii) otherwise, the term is stemmed and then matched using an efficient exact string matching method [75] to the concept index.

We give a brief description of the various methods we experimented with below. We note that

[3] https://github.com/XI-lab/tag-recommendation-data-iswc2012

[4] We consider that the term is an acronym if it is ≤ 5 letters, all capitalized, and if we cannot find it in the Ubuntu corpus of American words [http://packages.ubuntu.com/lucid/wamerican]

each of the following methods was carefully examined and optimized to yield the best possible results we could get after series of tests (e.g., we use fined-grained document frequencies and optimal thresholds for all the methods below).

`tf`**:** Our first approach simply ranks potential tags by counting the number of matches between the terms appearing in the paper and the concept index. While basic, this approach performs relatively well in our context since we consider a restricted number of terms only (our matching process is *mediated* through the ontology). In a standard setting without a field-specific ontology, this approach would perform poorly[5].

`tfidf`**:** This second method extends the approach above by applying standard TF-IDF [127]. We use a fine-grained document frequency in this case, based on the top categories of papers in arXiv rather than the entire document collection (i.e., IDF is computed based on the papers that share the same arXiv topic as the paper being bookmarked), as this performs better in practice.

`tf_simpleIDF`**:** In the ScienceWISE ontology, some scientific concepts are marked as "basic". While legitimate, those concepts are deemed rather general by our users and non-specific to any domain (*mass*, or *velocity* are two examples of such concepts). Under the `simpleIDF` scheme, IDF is not computed; rather, the system simply penalizes basic concepts and systematically puts them at the bottom of the ranked list (i.e., the ranked list of basic tags appears after the ranked list of other tags).

`tfidf_title`**:** The scientific terms that appear in titles and abstracts of the scientific papers often carry some special significance. Hence, we modify the TF-IDF ranking to promote the concepts appearing in the title into the top positions of the ranking list. Along similar lines, any concept appearing in the abstract has its TF score doubled (which also promotes it higher up in the list).

`tf_title`**:** The same as above, but discarding IDF and only taking into account TF when ranking.

`combined`**:** In this approach we combine `tfidf_title` but use `simpleIDF` to compute the document frequency. As we will see below, it only marginally impacts the effectiveness of the approach while drastically reducing computational complexity for large collections of papers. This is the ranking method that we have decided to deploy on our current production version of ScienceWISE.

`ont-depth`**:** Scientific concepts are often organized hierarchically in our ontology, with more specific, sub-concepts deriving from higher-level more general concepts. In this approach, we try to penalize more general concepts (that have a smaller depth in the ontology) and favor more specific concepts. More specifically, we penalize more generic concept by $c\_depth/distance\_from\_root\_concept$ where c_depth is a constant (we use $c\_depth = 1$ below, which yields the best results in our setting).

[5] it would lead to a MAP smaller than 1% in our case

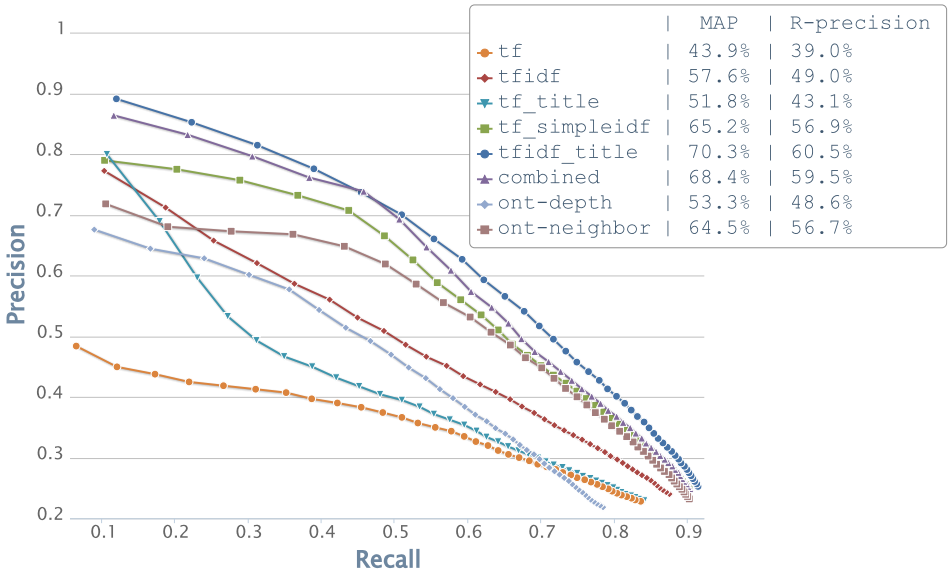

Figure 6.1 – Precision - Recall for our various tag recommendation approaches

`ont-neighbor`**:** Many scientific concepts are linked to further, related concepts in our ontology. Hence, we take advantage of the semantic graph connecting the concepts by improving the scores of those concepts that are direct neighbors of top-k ranked concepts. More specifically, we bump the ranking of direct neighbors of top-ranked concepts by $+c\_neighbor$ (we use $c\_neighbor = 3$ below, which yields the best results in our setting).

Figure 6.1 compares our different approaches on a Precision vs Recall graph along with the overall results in terms of MAP and R-precision. Results for Precision@k are depicted on Figure 6.2.

We observe the following:

1. Simple TF ranking yields the worst precision. However, a relatively minor improvement (boosting rank of concepts that occur in the title and abstract, `tf_title`) greatly improves performance for low k.

2. Performance of the `tfidf_title` is only marginally better than `combined`, with the latter one also being considerably faster (since the global IDF measure does not have to be computed). Both significantly outperform the standard `tfidf` ranking, which demonstrates that one can leverage the structure of scientific texts (where terms in the title and abstract are often very carefully chosen) in order to extract meaningful information.

3. The method leveraging the subsumption relations (*ont-depth*) performs surprisingly poorly. Further variants leveraging the subsumption hierarchies we experimented with behaved even worse. Choosing the right level in the hierarchy seems to be key, and hence favoring too

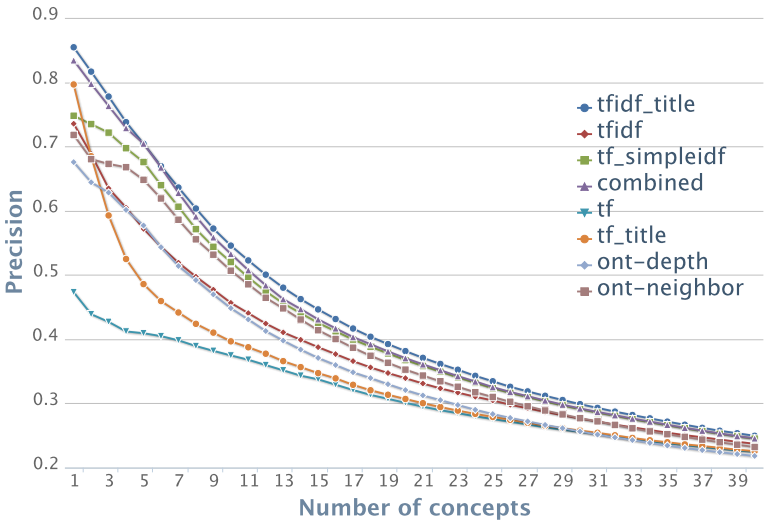

Figure 6.2 – Precision@k of our various ranking techniques for tag recommendation

specific (or, conversely, too generic) concepts yields suboptimal results (that are either too specific, and thus unrelated to the paper being analyzed, or too generic and thus are deemed less relevant also).

4. The method based on concept neighborhood (*ont-neighbor*) performs relatively well but is not better than simpler methods. The problem in that case seems to lie in the semantics of the relations between the concepts, which are often arbitrary in our ScienceWISE ontology and hence interconnect semantically heterogeneous concepts. One way of correcting this would be to (automatically or manually) create additional *same-as* or *see-also* relationships in our ontology, and to leverage such relationships to return additional relevant results (we successfully applied such techniques recently on the LOD graph, see [139]).

In summary, the careful use of some specific properties of the ontology (e.g., *basic* concepts) together with information about position of the terms in the document (e.g., in the title or abstract) allows to significantly increase precision in comparison with the baseline methods (increasing MAP up to 70%).

6.5.2 Disambiguating Tags

In order to tackle our second problem, we have implemented a special interface, that permits users to confirm or provide a disambiguation for acronyms and ambiguous concepts when bookmarking a paper. To help the user in this task, we cluster the collection of bookmarked papers into *topics* in an attempt to guess the correct disambiguation. We start by experimenting with the following techniques:

`lda`**:** Dirichlet Allocation (LDA) [18] is a standard tool in probabilistic topic modeling. Applied to IR, LDA basically considers that each document is a mixture of a small number of topics and that each word is attributable to one of the topics. It is conceptually similar to probabilistic Latent Semantic Analysis, except that in LDA the topic distributions are assumed to have Dirichlet priors, which often leads to better results in practice. We have used the LDA implementation from the Mallet package[6] in our experiments.

`k-means`**:** works similarly but takes advantage of the well-known k-means clustering technique to cluster the documents.

We consider our data set comprising papers from several disjoint arXiv subject classes[7] and split these collections into clusters using LDA and K-Means algorithms. The number of clusters is chosen to be equal to those of primary arXiv subject classes. After clustering, each concept is assigned to a unique cluster, and each paper is annotated with one or more clusters with different probabilities. This way we can rank concepts that disambiguate terms according to the cluster probability of the paper.

The resulting accuracy of LDA-based disambiguation is impressive (**75%**). One can in addition add ontological information to improve the disambiguation process and further boost the accuracy. For example, if among the concepts to disambiguate there is both a concept and subconcept (e.g. *power spectrum* and *matter power spectrum*) and if we provide the most specific concept, the accuracy raises to **88%**. We compare this to the standard k-means clustering algorithm, which only yields an accuracy of **47%**.

We evaluate our two disambiguation techniques on the test collection to see if they improve the effectiveness of tag recommendation. The Precision@k vs Recall@k graph is shown in Figure 6.3. On the graph, we additionally show the results of tag recommendation given perfect disambiguation ("GOLD disambiguation"). We observe that LDA-based disambiguation approach consistently outperforms the best approach without disambiguation ("tfidf_title").

Composite Concepts

Another approach to the disambiguation problem we experimented with is based on mereology and *composite concepts*. Concepts in a scientific ontology can often be expressed as *composites* of some other ontological concepts. For example, a concept *mass of particle* is a composite of two basic scientific concepts: *mass* and *particle*. Very often composite concepts are presented in many different textual forms. Moreover, it is custom to "shorten" the term (e.g. use *mass* instead of *mass of a star*, or simply *cluster* instead of *galaxy cluster*). Although this situation is formally similar to the previous one, it is impossible to guess what concepts should be disambiguated.

[6] http://mallet.cs.umass.edu/

[7] Each paper on arXiv belongs to one or several *Subject classes*, chosen by the authors of the paper

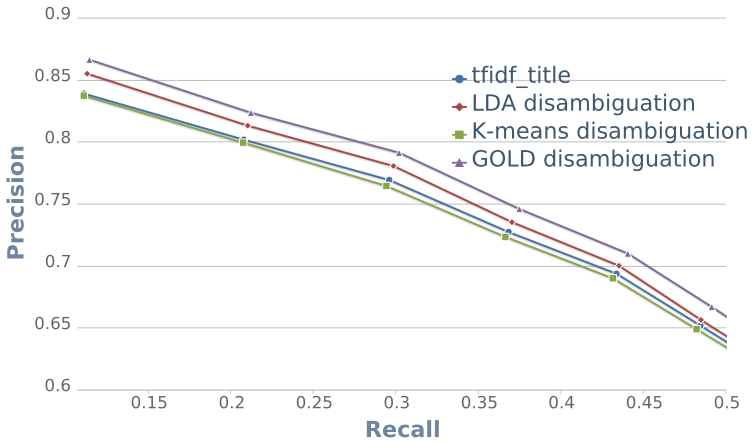

Figure 6.3 – Precision - Recall of tag recommendation with disambiguation

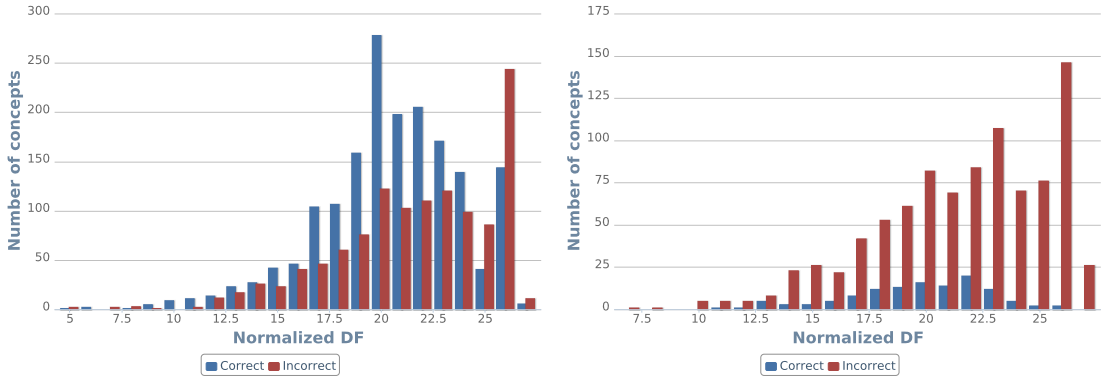

Figure 6.4 – Comparison of document frequency distribution for one-word concepts from the first 5 positions in the ranking (left panel) and from the positions (6–12). NormalizedDF is defined via Equation (6.1) in the text.

We have tested a hypothesis that *one-word concepts more often have a "generic meaning" than their many-words counterparts.*. If this is really the case, a proper tuning of the IDF function would be able to improve the ranking significantly. To determine whether this is indeed the case, we considered the *document frequency* (DF) distribution for the one-word concepts used as tags. The normalized DF on the x-axis is defined as

$$\text{normalized DF} = \log_{1.5}\left(\frac{\text{number of docs. containing a concept}}{\text{total number of docs. in collection}} \times 10^5\right) \tag{6.1}$$

The corresponding histograms are shown in Figure 6.4 where one can see (quite surprisingly) that the DF distribution for "correct" and "incorrect" concepts are roughly the same (although the correct ones are shifted somewhat to the lower DF region). Therefore, the one-word concepts bear no clear correlation with the document frequency. Based on these results, we

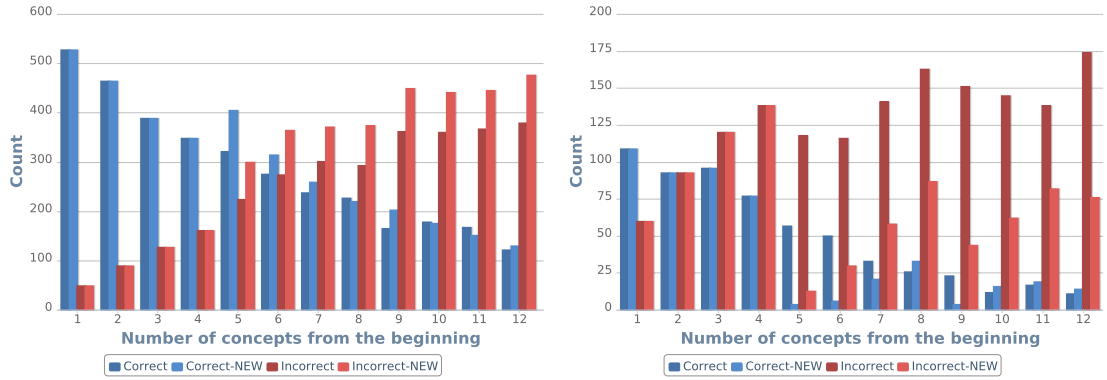

Figure 6.5 – Comparison of acceptance/rejection rate as a function of position in the ranking list before and after penalization of one-word concepts. Left panel shows change of the rejection rate for all concepts, right panel demonstrates rejection rate for one-word concepts.

decided to implement a simple strategy that penalizes one-word concepts that appear in positions 6 and below in our `tf_baseline` ranking. The results of this experiment are shown in Figure 6.5. Applied on our tag recommendation strategy, such a disambiguation approach yields and improvement in MAP of about 0.5% on average.

6.6 Conclusions

In this chapter, we addressed the problem of ontology-based tagging of scientific papers. We compared the effectiveness of various methods to recommend and disambiguate tags within a large-scale information system. Compared to classic tag recommendation, the proposed techniques select tags directly from a collaborative, user-driven ontology. Extensive experiments have shown that the use of a community-authored ontology together with information about the position of the concepts in the documents allows to significantly increase precision over standard methods. Also, several more specific techniques such as ontology-based neighborhood selection, LDA classification and one-word-concept penalization for tag disambiguation yield surprisingly good results and collectively represent a good basis for further experimentation and optimizations.

7 Conclusions

In this thesis, we investigated, designed, and evaluated a number methods and algorithms to mine structured data from unstructured sources. We made a series of contributions in Named Entity Recognition, Entity Disambiguation and Coreference Resolution. These tasks represent some of the core tasks of modern *Knowledge Extraction systems*. In particular, we explored Named Entity Recognition and Entity Disambiguation in the domain of idiosyncratic data. Additionally, we presented a next-generation system for scientific literature discovery and organization, ScienceWISE, whose development we also participated in. We believe that the approaches we introduced in this thesis contribute to the construction of high-quality Knowledge Graphs.

We started by tackling the problem of Named Entity Recognition in idiosyncratic domains in Chapter 3, motivated by the low quality of traditional NER systems on domain-specific data. We leveraged n-gram statistics over a collection of scientific papers on related topics to extract candidate named entities and then applied machine learning techniques to distill high-quality named entities from the candidates. Next, we proposed novel approaches for entity disambiguation in Chapter 4 that exploit both collection statistics as well as entity relations from the background ontology graph. We thoroughly evaluated and compared our approaches to more state-of-the-art techniques on two datasets taken from different scientific areas.

Subsequently, we turned our attention to the coreference resolution problem. In Chapter 5 we designed a new system, which added a semantic layer to a state-of-the-art coreference resolution system. Our method exploited semantic similarities of co-referring mentions and re-grouped them accordingly, which allowed us to outperform the state-of-the-art baseline.

Finally, in Chapter 6 we presented a next-generation system for discovery and organization of scientific literature, ScienceWISE. Discovery of new papers is one of the most important functionalities of the system. ScienceWISE allows to discover new articles based on collections of entities extracted from other relevant papers. Another feature of the system is paper organization via entity-centric tagging. In Chapter 6, we explored different strategies to

improve the tagging process and evaluated them on a collection taken directly from users interacting with the system.

7.1 Future Work

There are many research directions that are worth investigating in order to improve and develop new Knowledge Extraction and Discovery systems. In the following, we present some compelling ideas that could be pursued as an extension of this work, together with ideas that would require new platforms and knowledge extraction mechanisms.

7.1.1 Towards Integrated Text-to-Knowledge-Graph Platforms

The solutions to Knowledge Extraction tasks studied throughout this thesis were designed individually; Combined, they can form the basis of a novel knowledge extraction platform that offers modular mining of structured data from unstructured sources. Modularity will allow such a platform to activate individual solutions flexibly and create pipelines depending on the application at hand.

Furthermore, it is now possible to construct large of probabilistic graphs similar to the Knowledge Vault [33] directly from unstructured data. While this process can be seen as a one-off procedure, it can also benefit from a continuous iterative approach that updates the probabilistic knowledge graph every time new data arrives or there is a change in existing data and/or extraction algorithms. At the same time, a continuous extraction approach raises efficiency questions. The changes in the underlying data or the algorithms will require the knowledge graph to be updated, but re-running the complete extraction pipeline on a large-scale dataset from scratch could be time-consuming. We expect these issues to foster research on various optimizations on how to handle partial data changes in the most efficient manner.

7.1.2 Multi-Domain Knowledge Graphs

As discussed in Chapter 3, our machine learning classifier for candidate entities inferred different predictive features for documents coming from different idiosyncratic domains. In fact, we had to train classifiers individually for each collection to achieve the best results. Thus, robust performance of cross-domain Knowledge Extraction still requires additional improvements. Potentially, we need some more elaborated features that will work across the domains and a seed list of domain-independent extraction patterns as in [36].

Many attempts have been made to build knowledge graphs focused on molecular biology, such as genes and proteins, to mine interactions between proteins [58, 113]. Recently, there has been a larger effort to create a one-stop portal for a much wider range of entities and relations for life sciences. The proposed system, called KnowLife [35], uses seed facts and relational patterns to mine information about diseases, symptoms, drugs, side effects, and

more. It uses unstructured data from Web Portals in addition to scientific literature for its extraction mechanism, which is based on a small set of seed facts.

Medical knowledge also became a big market opportunity for major commercial players such as IBM and Google. Google, for example, in partnership with Mayo Clinic, started to populate its Knowledge Graph with healthcare decease data such as common symptoms, treatments, typical age group affected by the condition[1]. On the other side, IBM's Watson Health[2] targets doctors and healthcare providers to provide a decision support system based on their Watson technology. We expect the trend of fusing multi-domain data into knowledge graphs to continue and expect more domain-agnostic knowledge extraction methods to be developed.

7.1.3 Entity Disambiguation and Linking

In Chapter 4, we described and evaluated several approaches for Entity Disambiguation in relatively long, properly written texts, like abstracts of scientific papers in our case. As we saw, these texts typically contained 4 to 5 entity mentions, often more, which allowed us to disambiguate entity mentions based on other entities. Short textual content is a new frontier for modern Entity Linking, and is booming on various microblogging platforms, such as Twitter and Tumblr. Twitter is particularly interesting for many organizations, as it allows to monitor various events at a global scale. Posts in Twitter rarely contain more than one entity in it, thus entity disambiguation methods that rely on other entities in the text need to find new ways to disambiguate them. Moreover, due to the limited length of the tweets, entity labels also get shortened to new previously unseen forms. As a consequence, we need new methods to effectively identify entities in short textual documents; for example, pre-grouping documents based on certain document attributes (e.g., hashtags or authors of tweets) could be explored in this context.

7.1.4 Crowdsourcing for Knowledge Acquisition

Although we did not directly discuss crowdsourcing in this thesis (as we were mostly interested in automated knowledge extraction methods), it can be used as a powerful instrument to extract very precise and high-quality information. Crowdsourcing is already extensively used in knowledge extraction for annotating data with ground-truth information, e.g., when annotating texts with entities [139, 138]. However, it can be also used to harvest additional knowledge at scale. Vaish *et al* [146], for instance, recently proposed a crowdsourcing methodology via short queries on mobile phones for tasks such as census of local human activity, rating stock photos, or extracting structured data from Wikipedia pages. This way, we think crowdsourcing could be used to complete information in Knowledge Graphs at scale; for example, via queries targeted to collect information about local entities, such as businesses and organizations.

[1] https://googleblog.blogspot.com/2015/02/health-info-knowledge-graph.html
[2] http://www.ibm.com/smarterplanet/us/en/ibmwatson/health/

7.2 Outlook

The advent of Knowledge Graphs represent a landmark in the way we work with and process unstructured data. Knowledge Graphs can power many downstream applications including deep interpretation of natural language, semantic search and big data analytics over uncertain contents [136, 148]. Personal, user-centric Knowledge Graphs can be constructed from voice interactions with an intelligent assistance system [81]. Another domain of interest that emerged from large-scale knowledge extraction is smart assistants, i.e., intelligent systems that aim to bring the necessary information or take a certain action for an end-user via natural language interactions. Apple Siri, Miscrosoft Cortana and Google Now are a few products that answer user queries by leveraging the companies' respective Knowledge Graphs. In the future, we expect knowledge extraction methods to be developed further and allow Knowledge Graphs to grow, both in the number of different entity properties they hold and in the knowledge domains they cover. At the same time, many entities and properties in Knowledge Graphs will be encoded in a probabilistic manner, and will constantly evolve and change over time to reflect the changes in the underlying data.

Bibliography

[1] K. Abdalgader and A. Skabar. Unsupervised similarity-based word sense disambiguation using context vectors and sentential word importance. *ACM Trans. Speech Lang. Process.*, 9(1):2:1–2:21, May 2012.

[2] K. Aberer, A. Boyarsky, P. Cudré-Mauroux, G. Demartini, and O. Ruchayskiy. An integrated socio-technical crowdsourcing platform for accelerating returns in escience. In *ISWC (Outrageous Ideas Track)*, 2011.

[3] K. Aberer, A. Boyarsky, P. Cudré-Mauroux, G. Demartini, and O. Ruchayskiy. ScienceWISE : a Web-based Interactive Semantic Platform for scientic collaboration. In *10th International Semantic Web Conference (ISWC 2011-Demo), Bonn, Germany*, pages 87–102, 2011.

[4] B. Adrian, L. Sauermann, and T. Roth-berghofer. Contag: A semantic tag recommendation system. In *Proceedings of ISemantics' 07*, pages 297–304. JUCS, 2007.

[5] A. Alasiry, M. Levene, and A. Poulovassilis. Detecting candidate named entities in search queries. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 1049–1050, New York, NY, USA, 2012. ACM.

[6] T. Althoff, X. L. Dong, K. Murphy, S. Alai, V. Dang, and W. Zhang. Timemachine: Timeline generation for knowledge-base entities. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 19–28, New York, NY, USA, 2015. ACM.

[7] P. M. Andersen, P. J. Hayes, A. K. Huettner, L. M. Schmandt, I. B. Nirenburg, and S. P. Weinstein. Automatic extraction of facts from press releases to generate news stories. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, pages 170–177, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[8] M. Ariel. *Accessing noun-phrase antecedents*. Routledge, 2014.

[9] A. Athar and S. Teufel. Detection of implicit citations for sentiment detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, ACL '12, pages 18–26, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[10] N. Bach and S. Badaskar. A review of relation extraction.

[11] A. Bagga and B. Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer, 1998.

[12] B. Baldwin. Cogniac: high precision coreference with limited knowledge and linguistic resources. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 38–45. Association for Computational Linguistics, 1997.

[13] M. Bansal and D. Klein. Coreference semantics from web features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 389–398, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[14] P. Basile, M. Degemmis, A. L. Gentile, P. Lops, and G. Semeraro. The jigsaw algorithm for word sense disambiguation and semantic indexing of documents. In R. Basili and M. T. Pazienza, editors, *AI*IA*, volume 4733 of *Lecture Notes in Computer Science*, pages 314–325. Springer, 2007.

[15] O. Bender, F. J. Och, and H. Ney. Maximum entropy models for named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 148–151, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[16] A. L. Berger and V. O. Mittal. Ocelot: a system for summarizing web pages. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 144–151, New York, NY, USA, 2000. ACM.

[17] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics*, 7(3):154–165, Sept. 2009.

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[19] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Proceedings of the 6th Workshop on Very Large Corpora*, pages 152–160, 1998.

[20] R. F. Bruce and J. M. Wiebe. Decomposable modeling in natural language processing. *Comput. Linguist.*, 25(2):195–207, June 1999.

[21] V. Bryl, C. Giuliano, L. Serafini, and K. Tymoshenko. Using background knowledge to support coreference resolution. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 759–764, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[22] A. Budura, S. Michel, P. Cudré-Mauroux, and K. Aberer. Neighborhood-based tag prediction. In L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, and E. Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 608–622. 2009.

[23] D. Chen and C. Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October 2014. Association for Computational Linguistics.

[24] X. Cheng and D. Roth. Relational Inference for Wikification. *Empirical Methods in Natural Language Processing*, pages 1787–1796, 2013.

[25] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, Nov. 2011.

[26] J. Cowie and W. Lehnert. Information extraction. *Commun. ACM*, 39(1):80–91, Jan. 1996.

[27] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[28] W. Daelemans, A. Van Den Bosch, and J. Zavrel. Forgetting exceptions is harmful in language learning. *Mach. Learn.*, 34(1-3):11–41, Feb. 1999.

[29] H.-J. Dai, C.-Y. Chen, C.-Y. Wu, P.-T. Lai, R. T.-H. Tsai, and W.-L. Hsu. Coreference resolution of medical concepts in discharge summaries by exploiting contextual information. *Journal of the American Medical Informatics Association*, 19(5):888–896, 2012.

[30] J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):pp. 1470–1480, 1972.

[31] L. Del Corro and R. Gemulla. ClausIE: Clause-Based Open Information Extraction. In *Proceedings of the 22nd International World Wide Web Conference (WWW 2013)*, Rio do Janeiro, Brazil, 2013. International World Wide Web Conferences Steering Committee (IW3C2), ACM.

[32] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 469–478, New York, NY, USA, 2012. ACM.

[33] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 601–610, New York, NY, USA, 2014. ACM.

[34] P. Elango. Coreference resolution: A survey. Technical report, University of Wisconsin, Madison, 2005.

[35] P. Ernst, C. Meng, A. Siu, and G. Weikum. Knowlife: A knowledge graph for health and life sciences. In *Proceedings of the 2014 IEEE 30th International Conference on Data Engineering (ICDE)*, pages 1254–1257, March 2014.

[36] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.*, 165(1):91–134, June 2005.

[37] J. Fan, A. Kalyanpur, D. Gondek, and D. A. Ferrucci. Automatic knowledge extraction from documents. *IBM Journal of Research and Development*, 56(3.4):5–1, 2012.

[38] C. Fellbaum. Wordnet. *Theory and Applications of Ontology: Computer Applications*, pages 231–243, 2010.

[39] Y. feng Lin, T. han Tsai, W. chi Chou, K. pin Wu, T. yi Sung, and W. lian Hsu. A maximum entropy approach to biomedical named entity recognition. In *Proceedings of the 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics*, pages 56–61, 2004.

[40] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[41] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and et al. Domain-specific keyphrase extraction. In *Proceedings of the 16th international joint conference on Artificial Intelligence*, pages 668–673. Morgan Kaufmann Publishers, 1999.

[42] K. Frantzi, S. Ananiadou, and H. Mima. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2):115–130, 2000.

[43] R. Gabbard, M. Freedman, and R. Weischedel. Coreference for learning to extract relations: Yes, virginia, coreference matters. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 288–293, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[44] A. Gangemi, A. G. Nuzzolese, V. Presutti, F. Draicchio, A. Musetti, and P. Ciancarini. Automatic typing of DBpedia entities. In *The Semantic Web–ISWC 2012*, pages 65–81. Springer, 2012.

[45] N. Ge, J. Hale, and E. Charniak. A statistical approach to anaphora resolution. In *Proceedings of the sixth workshop on very large corpora*, volume 71, 1998.

[46] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Mach. Learn.*, 63(1):3–42, Apr. 2006.

[47] R. Grishman. Information extraction: Capabilities and challenges, 2012.

[48] B. J. Grosz et al. The representation and use of focus in a system for understanding dialogs. In *IJCAI*, volume 67, page 76, 1977.

[49] B. J. Grosz, S. Weinstein, and A. K. Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225, 1995.

[50] J. Guo, G. Xu, X. Cheng, and H. Li. Named entity recognition in query. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 267–274, New York, NY, USA, 2009. ACM.

[51] Z. Guo and D. Barbosa. Robust entity linking via random walks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 499–508, New York, NY, USA, 2014. ACM.

[52] A. Haghighi and D. Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1152–1161, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[53] A. Haghighi and D. Klein. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 385–393, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[54] H. Hajishirzi, L. Zilles, D. S. Weld, and L. S. Zettlemoyer. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 289–299, 2013.

[55] M. A. Halliday and R. Hasan. *Cohesion in English*. Longman, London, 1976.

[56] X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *SIGIR*, pages 765–774, New York, NY, USA, 2011. ACM.

[57] S. M. Harabagiu, R. C. Bunescu, and S. J. Maiorano. Text and knowledge mining for coreference resolution. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics, 2001.

[58] N. Harmston, W. Filsell, and M. P. H. Stumpf. Which species is it? species-driven gene name disambiguation using random walks over a mixture of adjacency matrices. *Bioinformatics*, 28(2):254–260, Jan. 2012.

[59] J. Hobbs. Resolving pronoun references. In *Readings in natural language processing*, pages 339–352. Morgan Kaufmann Publishers Inc., 1986.

[60] J. Hoffart, S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. Kore: Keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 545–554, New York, NY, USA, 2012. ACM.

[61] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 782–792, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[62] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Bibsonomy: A social bookmark and publication sharing system. In *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, pages 87–102, 2006.

[63] N. Houlsby and M. Ciaramita. A scalable Gibbs sampler for probabilistic entity linking. In *Advances in Information Retrieval*, pages 335–346. Springer, 2014.

[64] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene. Unsupervised graph-based topic labelling using dbpedia. In *Proceedings of the 6th ACM international conference on Web Search and Data Mining*, WSDM '13, pages 465–474, New York, NY, USA, 2013. ACM.

[65] P. S. Jacobs and L. F. Rau. Scisor: Extracting information from on-line news. *Commun. ACM*, 33(11):88–97, Nov. 1990.

[66] R. Jäschke, L. B. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in social bookmarking systems. *AI Commun.*, 21(4):231–247, 2008.

[67] X. Jiang, Y. Hu, and H. Li. A ranking approach to keyphrase extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 756–757, New York, NY, USA, 2009. ACM.

[68] A. Jimeno Yepes and A. R. Aronson. Knowledge-based and knowledge-lean methods combined in unsupervised word sense disambiguation. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, IHI '12, pages 733–736, New York, NY, USA, 2012. ACM.

[69] A. J. Jimeno-Yepes, B. T. McInnes, and A. R. Aronson. Exploiting mesh indexing in medline to generate a data set for word sense disambiguation. *BMC Bioinformatics*, 12:223, 2011.

[70] P. Jindal and D. Roth. Using domain knowledge and domain-inspired discourse model for coreference resolution for clinical narratives. *Journal of the American Medical Informatics Association*, 20(2):356–362, 2013.

[71] S. R. Jonnalagadda, D. Li, S. Sohn, S. T.-I. Wu, K. Wagholikar, M. Torii, and H. Liu. Coreference analysis in clinical notes: a multi-pass sieve with alternate anaphora resolution modules. *Journal of the American Medical Informatics Association*, 19(5):867–874, 2012.

[72] J. Kazama and K. Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707, 2007.

[73] S. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin. Automatic keyphrase extraction from scientific articles. *Language Resources and Evaluation*, pages 1–20, 2012.

[74] D. Klein, J. Smarr, H. Nguyen, and C. D. Manning. Named entity recognition with character-level models. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 180–183, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[75] D. E. Knuth, J. J. H. Morris, and V. R. Pratt. Fast pattern matching in strings. *SIAM Journal on Computing*, 6(2):323–350, 1977.

[76] M. Krapivin, M. Autayeu, M. Marchese, E. Blanzieri, and N. Segata. Improving machine learning approaches for keyphrases extraction from scientific documents with natural language knowledge. In *Proceedings of the joint JCDL/ICADL international digital libraries conference*, pages 102–111, 2010.

[77] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 457–466, New York, NY, USA, 2009. ACM.

[78] S. Lappin and H. J. Leass. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561, 1994.

[79] Y. K. Lee and H. T. Ng. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *ACL-02 - Volume 10*, EMNLP '02, pages 41–48, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[80] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee. Twiner: named entity recognition in targeted twitter stream. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 721–730, New York, NY, USA, 2012. ACM.

[81] X. Li, G. Tur, D. Hakkani-Tür, and Q. Li. Personal knowledge graph population from user utterances in conversational understanding. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 224–229, Dec 2014.

[82] M. Lipczak. Tag recommendation for folksonomies oriented towards individual users. *ECML PKDD Discovery Challenge*, 2008.

[83] V. Lopez, M. Fernández, E. Motta, and N. Stieler. Poweraqua: Supporting users in querying and exploring the semantic web. *Semant. web*, 3(3):249–265, Aug. 2012.

[84] X. Luo. On coreference resolution performance metrics. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32. Association for Computational Linguistics, 2005.

[85] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[86] J. Mayfield, P. McNamee, and C. Piatko. Named entity recognition using hundreds of thousands of features. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 184–187, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[87] E. Meij, K. Balog, and D. Odijk. Entity linking and retrieval. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 1127–1127, New York, NY, USA, 2013. ACM.

[88] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 1–8, New York, NY, USA, 2011. ACM.

[89] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management*, CIKM '07, pages 233–242, New York, NY, USA, 2007. ACM.

[90] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[91] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

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

[93] G. Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In L. Carr, D. D. Roure, A. Iyengar, C. A. Goble, and M. Dahlin, editors, *WWW*, pages 953–954. ACM, 2006.

[94] R. Mitkov. Robust pronoun resolution with limited knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, pages 869–875. Association for Computational Linguistics, 1998.

[95] D. Mollá, M. V. Zaanen, and D. Smith. Named entity recognition for question answering. In *In Lawrence Cavedon and Ingrid Zukerman, editors, Proceedings of the 2006 Australasian Language Technology Workshop*, pages 51–58, 2006.

[96] A. Moro, A. Raganato, and R. Navigli. Entity linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.

[97] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[98] N. Nakashole, T. Tylenda, and G. Weikum. Fine-grained semantic typing of emerging entities. In *ACL (1)*, pages 1488–1497, 2013.

[99] R. Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69, Feb. 2009.

[100] R. Navigli, S. Faralli, A. Soroa, O. de Lacalle, and E. Agirre. Two birds with one stone: learning semantic models for text categorization and word sense disambiguation. In *CIKM*, pages 2317–2320, New York, NY, USA, 2011. ACM.

[101] V. Ng. Machine learning for coreference resolution: Recent successes and future challenges. Technical report, Cornell University, 2003.

[102] V. Ng. Semantic class induction and coreference resolution. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 536–543, 2007.

[103] V. Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1396–1411. Association for Computational Linguistics, 2010.

[104] H. Paulheim and C. Bizer. Improving the Quality of Linked Data Using Statistical Distributions. *Int. J. Semantic Web Inf. Syst.*, 10(2):63–86, Jan. 2014.

[105] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[106] G. Pirró. Reword: Semantic relatedness in the web of data. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 129–135. AAAI Press, 2012.

[107] J. Plu, G. Rizzo, and R. Troncy. *Semantic Web Evaluation Challenges: Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers*, chapter A Hybrid Approach for Entity Recognition and Linking, pages 28–39. Springer International Publishing, Cham, 2015.

[108] T. Poibeau and L. Kosseim. Proper name extraction from non-journalistic texts. In *Computational Linguistics in the Netherlands*, pages 144–157, 2001.

[109] S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 192–199. Association for Computational Linguistics, 2006.

[110] M. F. Porter. Readings in information retrieval. chapter An algorithm for suffix stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[111] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, CoNLL '12, pages 1–40, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[112] R. Prokofyev, G. Demartini, and P. Cudré-Mauroux. Effective named entity recognition for idiosyncratic web collections. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 397–408, New York, NY, USA, 2014. ACM.

[113] S. Pyysalo, T. Ohta, M. Miwa, H.-C. Cho, J. Tsujii, and S. Ananiadou. Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18):i575–i581, 2012.

[114] K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 492–501, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[115] A. Rahman and V. Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 968–977, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[116] A. Rahman and V. Ng. Coreference resolution with world knowledge. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 814–824, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[117] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[118] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the 13th conference on Computational Natural Language Learning (CONLL)*, pages 147–155, 2009.

[119] L. Ratinov and D. Roth. Learning-based multi-sieve co-reference resolution with knowledge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1234–1244, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[120] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1375–1384, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[121] A. Ratnaparkhi et al. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, pages 133–142, 1996.

[122] M. Recasens and E. Hovy. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17(4):485–510, Oct. 2011.

[123] G. Rizzo and T. Raphaël. NERD : A Framework for Evaluating Named Entity Recognition Tools in the Web of Data. *Proceedings of the 11th Interational Semantic Web Conference ISWC2011*, pages 1–4, 2011.

[124] P.-M. Ryu, M.-G. Jang, and H.-K. Kim. Open domain question answering using wikipedia-based knowledge model. *Inf. Process. Manage.*, 50(5):683–692, Sept. 2014.

[125] S. Sahoo, A. Sheth, and C. Henson. Semantic provenance for escience: Managing the deluge of scientific data. *Internet Computing, IEEE*, 12(4):46–54, 2008.

[126] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, Aug. 1988.

[127] G. Salton and M. J. McGill. Introduction to modern information retrieval. 1986.

[128] A. Saxena, A. Jain, O. Sener, A. Jami, D. K. Misra, and H. S. Koppula. Robobrain: Large-scale knowledge engine for robots. *CoRR*, abs/1412.0691, 2014.

[129] M. Schuhmacher and S. P. Ponzetto. Exploiting dbpedia for web search results clustering. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, pages 91–96, New York, NY, USA, 2013. ACM.

[130] B. Settles. ABNER: An open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, 21(14):3191–3192, 2005.

[131] Y. Shinyama and S. Sekine. Named entity discovery using comparable news articles. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[132] C. Sidner. Focusing in the comprehension of definite anaphora. In *Readings in Natural Language Processing*, pages 363–394. Morgan Kaufmann Publishers Inc., 1986.

[133] C. L. Sidner. Towards a computational theory of definite anaphora comprehension in english discourse. Technical report, DTIC Document, 1979.

[134] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544, 2001.

[135] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, volume 6, pages 1419–1424, 2006.

[136] F. Suchanek and G. Weikum. Knowledge harvesting in the big-data era. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 933–938, New York, NY, USA, 2013. ACM.

[137] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[138] A. Tonon, M. Catasta, G. Demartini, P. Cudré-Mauroux, and K. Aberer. Trank: Ranking entity types using the web of data. In *The Semantic Web – ISWC 2013*, volume 8218 of *Lecture Notes in Computer Science*, pages 640–656. Springer Berlin Heidelberg, 2013.

[139] A. Tonon, G. Demartini, and P. Cudré-Mauroux. Combining inverted indices and structured search for ad-hoc object retrieval. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 125–134, New York, NY, USA, 2012. ACM.

[140] A. Toral, E. Noguera, F. Llopis, and R. Muñoz. *Natural Language Processing and Information Systems: 10th International Conference on Applications of Natural Language to Information Systems, NLDB 2005, Alicante, Spain, June 15-17, 2005. Proceedings*, chapter Improving Question Answering Using Named Entity Recognition, pages 181–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[141] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[142] P. D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, May 2000.

[143] T. Tylenda, M. Sozio, and G. Weikum. Einstein: Physicist or vegetarian? summarizing semantic type graphs for knowledge discovery. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 273–276, New York, NY, USA, 2011. ACM.

[144] C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 639–648, New York, NY, USA, 2012. ACM.

[145] O. Uryupina, M. Poesio, C. Giuliano, and K. Tymoshenko. Disambiguation and filtering methods in using web knowledge for coreference resolution. In *FLAIRS Conference*, pages 317–322, 2011.

[146] R. Vaish, K. Wyngarden, J. Chen, B. Cheung, and M. S. Bernstein. Twitch crowdsourcing: Crowd contributions in short bursts of time. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 3645–3654, New York, NY, USA, 2014. ACM.

[147] K. Van Deemter and R. Kibble. On coreferring: Coreference in muc and related annotation schemes. *Computational linguistics*, 26(4):629–637, 2000.

[148] N. H. Vo. Answering deep queries specified in natural language with respect to a frame based knowledge base and developing related natural language understanding components, 2015. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2016-03-07.

[149] U. Waltinger, A. Breuing, and I. Wachsmuth. Interfacing virtual agents with collaborative knowledge: Open domain question answering using wikipedia-based topic models. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 1896–1902. AAAI Press, 2011.

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

[151] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Collaborative web tagging workshop at WWW2006, Edinburgh, Scotland*, 2006.

[152] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, NAACL-Demonstrations '07, pages 25–26, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[153] W. Zheng, H. Fang, and C. Yao. Exploiting concept hierarchy for result diversification. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1844–1848, New York, NY, USA, 2012. ACM.

[154] L. Zou, R. Huang, H. Wang, J. X. Yu, W. He, and D. Zhao. Natural language question answering over rdf: A graph data driven approach. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 313–324, New York, NY, USA, 2014. ACM.

Roman Prokofyev

1700 Fribourg, Switzerland
☎ +41 26 552 0004
✆ +41 76 529 7448
✉ roman.prokofyev@gmail.com

Currently I am a PhD student in CS working in areas of big data, semantic web and natural language processing. I particularly enjoy open-ended and research-oriented problems and favor tools and technologies that allow me to tackle these problems most effectively, such as functional programming languages and noSQL databases.

My interests also include mobile technologies, data analytics and open data.

Technical Skills

Programming Languages *Python, Java, JavaScript*

Technologies *scikit-learn, nltk, hadoop, spark, play framework, jquery, mongodb*

Education

Nov. 2011 – Present **PhD Student**, University of Fribourg.
PhD student at the eXascale Infolab, under supervision of Professor Philippe Cudré-Mauroux. My research is focused on developing new methods that provide effective knowledge extraction from unstructured technical data, knowledge processing and discovery. In particular, I worked on the following research problems:
- Tag recommendation for scientific papers based on the the ontology of scientific concepts and user's interest profile.
- Ontology-based word sense disambiguation for scientific literature.
- Named Entity Recognition for scientific literature.

2007–2009 **M.Sc. in Applied Mathematics and Physics**, *Moscow Institute of Physics and Technology*, Moscow, Russia.
Thesis: Building enterprise network monitoring and management system based on open source software (OpenNMS).
GPA: 5/5.

2003–2007 **B.Sc. in Applied Mathematics and Physics**, *Moscow Institute of Physics and Technology*, Moscow, Russia.
Thesis: Effectiveness analysis of Intrusion Detection Systems.
GPA: 4.7/5.

Experience

Feb. 2010 – Mar. 2011 **Python/Django Developer**, École Polytechnique Fédérale de Lausanne (EPFL).

My general responsibilities were to maintain and develop new features for a scientific application running on the Python/Django platform (ScienceWISE.info). The system is processing large amounts of scientific papers from the arxiv.org website on a daily basis. Every paper is automatically annotated using the concepts from the ScienceWISE Ontology. The links between articles and concepts are then used to provide a semantic search capabilities over collections of articles.

I have also implemented a number of sub-projects that expanded the functionality of the system, in particular:

- A full text search solution based on the Whoosh engine.
- A multi-purpose tool to extract taxonomies of scientific concepts from external resources, such as Wikipedia and other online encyclopaedias; and further integrate them to the ScienceWISE Ontology.
- An authentication system based on SWITCH AAI.
- A live LaTeX-editor for writing definitions of scientific concepts.

The paper presenting the project received the best demo paper award at International Semantic Web Conference 2011.

Dec. 2007 – Jan. 2010 **Infrastructure Engineer**, COS&HT.

My main responsibilities included implementing various infrastructure projects for customers as well as maintenance of company's internal infrastructure solutions. The stack of solutions was primarily based on the enterprise products of Sun Microsystems, such as Sun Java Identity Manager, Sun Secure Global Desktop, etc.
Noticeable customer projects:

- Lead engineer in implementing Identity Management solution for Russian Federal Customs Service based on Sun Java Identity Manager.
- Engineer in implementing infrastructure solutions for Russian regional administrations providing governmental and municipal services.

Jun. 2006 – Nov. 2007 **Software Engineer**, NetCracker, System Performance dept.

Our department was dedicated to investigating and solving complex issues with unusually slow performance and memory leaks in J2EE applications. This work required performing lots of complex debugging and analysis of Java heapdumps.

During my work, I have also developed a *Jython* application for automated deployment of J2EE applications on *WebLogic Application Server*.

Additional Projects

May 2012 – Present **FarPlano**.

Android app for public transportation in Switzerland.
play.google.com/store/apps/details?id=com.schedulr

Sep. 2011 – Present **django-selenium**, *pypi.python.org/pypi/django-selenium*.

Selenium testing library integration for Django Framework.