Exploratory Relation Extraction in Large Multilingual Data

vorgelegt von
M.Sc.
Alan Akbik
geb. in Oberhausen

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Klaus-Robert Müller

Gutachter: Prof. Dr. Volker Markl
Gutachter: Prof. Dr. Hans Uszkoreit
Gutachter: Prof. Dr. Chris Biemann

Tag der wissenschaftlichen Aussprache: 13. April 2016

Berlin 2016

TECHNISCHE UNIVERSITÄT BERLIN

DOCTORAL THESIS

Exploratory Relation Extraction in Large Multilingual Data

Author:
Alan AKBIK

Supervisor:
Prof. Dr. Volker MARKL

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Engineering

in the

Database Systems and Information Management Group
Institute of Software Engineering and Theoretical Computer Science

Berlin, 2016

Declaration of Authorship

I, Alan AKBIK, declare that this thesis titled, 'Exploratory Relation Extraction in Large Multilingual Data' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ______________________________

Date: ______________________________

Doubt. *In a sense, that had been Achamian's single lesson. Geometry, logic, history, mathematics using Nilnameshi numbers, even philosophy!—all these things were dross, Achamian would argue, in the face of doubt. Doubt had made them, and doubt would unmake them.*
Doubt, he would say, set men free . . . Doubt, not truth*!*
Beliefs were the foundation of actions. Those who believed without doubting, he would say, acted without thinking. And those who acted without thinking were enslaved.
That was what Achamian would say.

– Excerpt from [1]

TECHNISCHE UNIVERSITÄT BERLIN

Abstract

Database Systems and Information Management Group
Institute of Software Engineering and Theoretical Computer Science

Doctor of Engineering

Exploratory Relation Extraction in Large Multilingual Data

by Alan AKBIK

The task of Relation Extraction (RE) is concerned with creating extractors that automatically find structured, relational information in unstructured data such as natural language text. Motivated by an explosion of sources of readily available text data such as the web, RE offers intriguing possibilities for querying, organizing, and analyzing information by drawing upon the clean semantics of structured databases and the abundance of unstructured data. However, practical applications of RE are often characterized by vague and shifting information needs on the one hand and large multilingual datasets of unknown content on the other. Classical RE approaches are unable to handle such scenarios since they require a careful, upfront definition of extraction tasks before extractors can be created in an effort-intensive, time-consuming process.

With this thesis, I propose the paradigm of Exploratory Relation Extraction (ERE), a user-driven but data-guided process of *exploration* for relations of interest in unknown data. I show how distributional evidence and an informed linguistic abstraction can be employed to allow users to openly explore a dataset for relations of interest and rapidly prototype extractors for discovered relations at minimal effort. Furthermore, I propose the use of a *language-neutral representation of shallow semantics* to address the issue of multilingual data. This representation enables a shared feature space for different languages against which extractors can be developed. I present a method that expands English-language Semantic Role Labeling (SRL) to other languages and use it to generate multilingual SRL resources for seven distinct languages from different language groups, namely Arabic, Chinese, French, German, Hindi, Russian and Spanish in order to bootstrap semantic parsers for these languages. Together, the researched approaches represent a novel way for data scientists to work with large multilingual datasets of unknown content.

TECHNISCHE UNIVERSITÄT BERLIN

Kurzfassung

Fachgebiet Datenbanksysteme und Informationsmanagement
Institut für Softwaretechnik und Theoretische Informatik

Doktor der Ingenieurwissenschaften

Exploratory Relation Extraction in Large Multilingual Data

by Alan AKBIK

Die Problemstellung der Relationsextraktion (RE) beschreibt die automatische Gewinnung strukturierter, relationaler Information aus unstrukturierten Daten wie zum Beispiel natürlichsprachlichem Text. Durch RE werden neue Arten der Strukturierung, Organisation und Analyse von Informationen ermöglicht, da sie eine Brücke zwischen der klar strukturierten Semantik von Datenbanken und der stetigen Explosion verfügbarer Textdaten zu bauen vermag. In der Praxis ist die Anwendung von RE allerdings problematisch; Anwendungsszenarien sind oft durch vage, sich schnell ändernde Informationsbedürfnisse gekennzeichnet, sowie von großen, mehrsprachigen Datensätzen unbekannten Inhalts. In solchen Szenarien schlagen klassische RE Ansätze fehl, da Extraktionsaufgaben im Voraus sorgsam definiert werden müssen, woraufhin Extraktoren in einem zweiten Schritt mit hohem Aufwand gebaut werden.

In dieser Dissertation stelle ich das neuartige Paradigma der Explorativen Relationsextraktion (ERE) vor. Hierbei handelt es sich um einen nutzergetriebenen, halbautomatischen Vorgang, mit dem neue Relationstypen in Datensätzen unbekannten Inhalts entdeckt werden können. Ich zeige, wie verteilungssemantische Statistiken und eine ausgewählte linguistische Abstraktion angewendet werden, um Nutzern sowohl die Erkundung von Textdaten nach relationalen Informationen als auch das schnelle prototypische Erstellen von Extraktoren mit minimalem Aufwand zu ermöglichen. Für den Umgang mit mehrsprachigen Daten schlage ich darüber hinaus die Nutzung einer sprachübergreifenden Repräsentation flacher Semantik vor. Auf dieser Basis können ohne Zusatzaufwand sprachübergreifende Extraktoren erzeugt werden. Ich stelle eine Methode vor, mit der englischsprachige Semantische Rollen auf andere Sprachen ausgeweitet werden können und erzeuge damit umfassende Resourcen, um mehrsprachige semantische Parser zu trainieren. Zusammengenommen stellen die in dieser Dissertation erforschten Methoden einen neuartigen Ansatz zum Umgang mit großen und mehrsprachigen Datensätzen unbekannten Inhalts dar.

Acknowledgements

First and foremost, I would like to thank my advisor Prof. Volker Markl for giving me the opportunity to follow my passion and carry out this research. This work would not have been possible without his persistent support and the great research environment he created. I would also like to thank Prof. Hans Uszkoreit for detailed and insightful discussions of this research, as well as Prof. Chris Biemann for his support and many inspiring discussions.

Furthermore, I wish to thank my team at IBM Research - Shivakumar Vaithyanathan, Yunyao Li, Laura Chiticariu, Marina Danilevsky and Huaiyu Zhu - for an unforgettable experience at Almaden. I also thank my research comrade Larysa Visengeriyeva, my office buddies Christoph Boden and Sebastian Schelter, my partner-in-crime Martin Schenck, my fellow pipeline tinkerers Johannes Kirschnick and Torsten Kilias, my students Umar Maqsud, Stefan Schramm, Do Tuan Anh and Thilo Michael, my proof readers Reda Al Azmeh, Sergey Dudalov, Moritz Schubotz and Marcus Leich, and all other colleagues and students I have been fortunate to work with at DIMA. I count myself lucky to have worked with and befriended so many great people over the past four years.

Contents

Declaration of Authorship i

Opening Quote ii

Abstract iii

Kurzzusammenfassung iv

Acknowledgements v

Contents vi

List of Figures x

List of Tables xi

Abbreviations xii

1 Introduction 1
1.1 Motivation and Problem Statement 1
1.2 Main Contributions 5
1.3 Thesis Structure 8

2 Preliminaries 9
2.1 Overview 9
2.2 Relation Extraction 10
2.2.1 Early History 10
2.2.1.1 MUC-1 and MUC-2: First Steps 10
2.2.1.2 MUC-3 and MUC-4: Evaluation Measures 12
2.2.1.3 MUC-5: Multilingual Data 12
2.2.1.4 MUC-6 and MUC-7: Information Extraction Subtasks 13
2.2.1.5 ACE: Refinement of Relation Extraction Task 14
2.2.1.6 Summary 15
2.2.2 Evaluation Measures 16

2.2.3 Methods . . . 17
2.2.3.1 Rule-Based . . . 18
2.2.3.2 Machine Learning-Based . . . 20
2.2.4 Summary and Discussion . . . 22
2.3 Lexical, Syntactic and Semantic Features . . . 23
2.3.1 Shallow Features . . . 24
2.3.2 Dependency Trees . . . 25
2.3.3 Semantic Role Labels . . . 26
2.3.3.1 Resources . . . 28
2.4 Summary of Terminology and Notation . . . 29

3 Relation Discovery **31**
3.1 Overview . . . 31
3.1.1 Problem Statement . . . 32
3.1.2 Related Work . . . 35
3.1.2.1 Pattern Generation for Relation Discovery . . . 35
3.1.2.2 Distributional Semantics for Other Tasks . . . 36
3.1.3 Contributions . . . 38
3.2 Pattern Generation for Relation Discovery . . . 38
3.2.1 Lexico-Syntactic Patterns . . . 39
3.2.1.1 Extended Shortest Path (PROPOSED-ESP) . . . 39
3.2.1.2 All Subtrees (PROPOSED-SUB) . . . 40
3.2.2 Entity Type Restrictions . . . 41
3.2.2.1 Named Entity Type Restrictions . . . 42
3.2.2.2 Fine-grained Entity Type Restrictions . . . 42
3.2.2.3 Phrasal Clusters as Restrictions . . . 44
3.3 Evaluation . . . 45
3.3.1 Experimental Setup . . . 45
3.3.1.1 Dataset . . . 45
3.3.1.2 Pattern Generation . . . 47
3.3.1.3 Feature Weighting . . . 49
3.3.1.4 Clustering Approach and Parameterization . . . 50
3.3.1.5 Evaluation Measures . . . 50
3.3.2 Experiment 1: k-Means at Fixed k . . . 52
3.3.2.1 Quantitative Evaluation . . . 52
3.3.2.2 Qualitative Analysis . . . 55
3.3.3 Experiment 2: k-Means at Variable k . . . 57
3.3.3.1 Quantitative Evaluation . . . 58
3.3.3.2 Qualitative Analysis . . . 58
3.4 Discussion and Summary . . . 59

4 Exploratory Relation Extraction **61**
4.1 Overview . . . 61
4.1.1 Problem Statement . . . 62
4.1.2 Previous Work . . . 63
4.1.3 Contributions . . . 65
4.2 Exploratory Relation Extraction . . . 65

4.2.1 Data-Guided Exploratory Workflow . . . 66
4.2.1.1 Overview . . . 66
4.2.1.2 Example 1: Vague Information Need . . . 67
4.2.1.3 Example 2: Exploratory Search for Relations . . . 69
4.2.2 Exploratory Pattern Queries . . . 70
4.2.2.1 Readability . . . 70
4.2.2.2 Preemptive Extraction and Indexing . . . 71
4.2.3 The SCHNÄPPER Toolkit . . . 73
4.2.3.1 Web Interface . . . 73
4.2.3.2 Example Usage . . . 75
4.3 Evaluation . . . 76
4.3.1 Experimental Setup . . . 76
4.3.1.1 Dataset . . . 76
4.3.1.2 User Preparation . . . 77
4.3.1.3 Evaluation Tasks . . . 77
4.3.2 Experiment 1: Extraction Task . . . 78
4.3.2.1 Quantitative Evaluation . . . 78
4.3.2.2 Qualitative Analysis . . . 79
4.3.3 Experiment 2: Exploration Task . . . 80
4.3.3.1 Qualitative Analysis . . . 81
4.4 Discussion and Summary . . . 82

5 Multilingual Semantic Role Labeling 84
5.1 Overview . . . 84
5.1.1 Problem Statement . . . 85
5.1.2 Previous Work . . . 86
5.1.3 Contributions . . . 88
5.2 Generating Multilingual SRL Resources . . . 90
5.2.1 Stage 1: Filtered Annotation Projection . . . 90
5.2.1.1 Analysis of Direct Projection . . . 91
5.2.1.2 Filtering Policies . . . 93
5.2.1.3 Effectiveness of Filtering Policies . . . 94
5.2.2 Stage 2: Bootstrapped Training of SRL . . . 95
5.2.2.1 Analysis of Relabeling Approach . . . 95
5.2.2.2 Bootstrap Learning . . . 96
5.2.2.3 Effectiveness of Bootstrapping . . . 97
5.3 Evaluation . . . 98
5.3.1 Experimental Setup . . . 98
5.3.1.1 Resources for each TL . . . 99
5.3.1.2 Evaluation Task . . . 99
5.3.2 Experiment 1: Quality of Generated PropBanks . . . 101
5.3.3 Experiment 2: Impact of Additional Factors . . . 102
5.4 Discussion and Summary . . . 104

6 Conclusion 106
6.1 Summary . . . 106
6.2 Outlook . . . 107

Bibliography **108**

List of Figures

1.1 Overview Relation Extraction for analytics. 3

2.1 Example scenario template filling task from MUC-2. 11
2.2 Example entity template relations from MUC-7. 13
2.3 Main view of PROPMINER. 18
2.4 Results view of PROPMINER. 19
2.5 Schematic outline of bootstrapping approach. 21
2.6 Regular expression of POS tags used to validate between-text for RE. 23
2.7 Dependency parse of an example sentence. 25
2.8 Diathesis alternation correctly labeled using SRL. 27

3.1 Example of a pair-pattern matrix. 33
3.2 Dependency parse of an example sentence, with an entity pair its shortest path. 40
3.3 Pattern generation process for one example sentence. 42
3.4 Pattern generation with YAGO type restrictions. 43
3.5 Pattern generation with phrasal clusters as type restrictions. 44
3.6 Clustering quality in terms of precision, recall and F_1-measure on the R20-ZIPFIAN dataset. 57

4.1 Example for Exploratory Relation Extraction process. 68
4.2 Illustration of the ERE example workflow discussed in Section 4.2.1.3. 69
4.3 Illustration of the subtree generation process. 71
4.4 Screen capture of the SCHNÄPPER tool. 73

5.1 Illustration of semantic roles across languages. 85
5.2 Example sentence pair for annotation projection. 87
5.3 Overview of proposed annotation projection approach. 89
5.4 Illustration of direct projection. 90
5.5 Values at each bootstrap iteration. 98
5.6 Screenshot of the evaluation interface. 100

List of Tables

2.1 Overview over MUC tasks. 15
2.2 Contingency table for evaluating Relation Extraction. 16

3.1 Sentences with different pattern generation methods. 34
3.2 Pattern generated with different methods. 41
3.3 Distant supervision example sentences. 46
3.4 The four evaluation datasets created using YAGO and distant supervision. . . . 47
3.5 Evaluation of different pattern generation methods for the two R20 datasets. . . 53
3.6 Evaluation of different pattern generation methods for the two R10 datasets. . . 54
3.7 Example clusters found with different pattern generation setups. 55

4.1 Evaluation results of Exploratory Relation Extraction for the extraction task. . . 78
4.2 Qualitative analysis of error sources for Exploratory Relation Extraction 79
4.3 Exaples for relations discovered with Exploratory Relation Extraction in the exporation task. 81

5.1 Breakdown of error classes in predicate projection. 92
5.2 Breakdown of error classes in argument projection. 92
5.3 Quality of predicate and argument labels for different projection methods on $\texttt{French}_{\text{gold}}$, including upper bound. 94
5.4 Experimental setup of annotation projection. 99
5.5 Estimated precision and recall over seven languages. 101
5.6 Estimated impact of downsampling parallel corpus. 102
5.7 Impact of additional factors. 102
5.8 Characteristics of the generated PROPBANKS. 103

Abbreviations

ACE	**A**utomatic **C**ontent **E**xtraction
ERE	**E**xploratory **R**elation **E**xtraction
ES	**E**xploratory **S**earch
IE	**I**nformation **E**xtraction
KB	**K**nowledge **B**ase
ML	**M**achine **L**earning
MUC	**M**essage **U**nderstanding **C**onference
NE	**N**amed **E**ntity
NER	**N**amed **E**ntity **R**ecognition
NLP	**N**atural **L**anguage **P**rocessing
OpenIE	**Open** **I**nformation **E**xtraction
PMI	**P**ointwise **M**utual **I**nformation
RD	**R**elation **D**iscovery
RE	**R**elation **E**xtraction
SMT	**S**tatistical **M**achine **T**ranslation
SL	**S**ource **L**anguage
SRL	**S**emantic **R**ole **L**abeling
TL	**T**arget **L**anguage
URE	**U**nsupervised **R**elation **E**xtraction

To my parents

1

Introduction

"And just what comes before?" Cnaiür asked, trying to force a sneer.

"For Men? History. Language. *Passion. Custom. All these things determine what men say, think, and do. These are the hidden puppet-strings from which all men hang."*

Shallow breath. A face freighted by unwanted insights.

"And when the strings are seen. . . "

"They may be seized."

– Excerpt from [2]

1.1 Motivation and Problem Statement

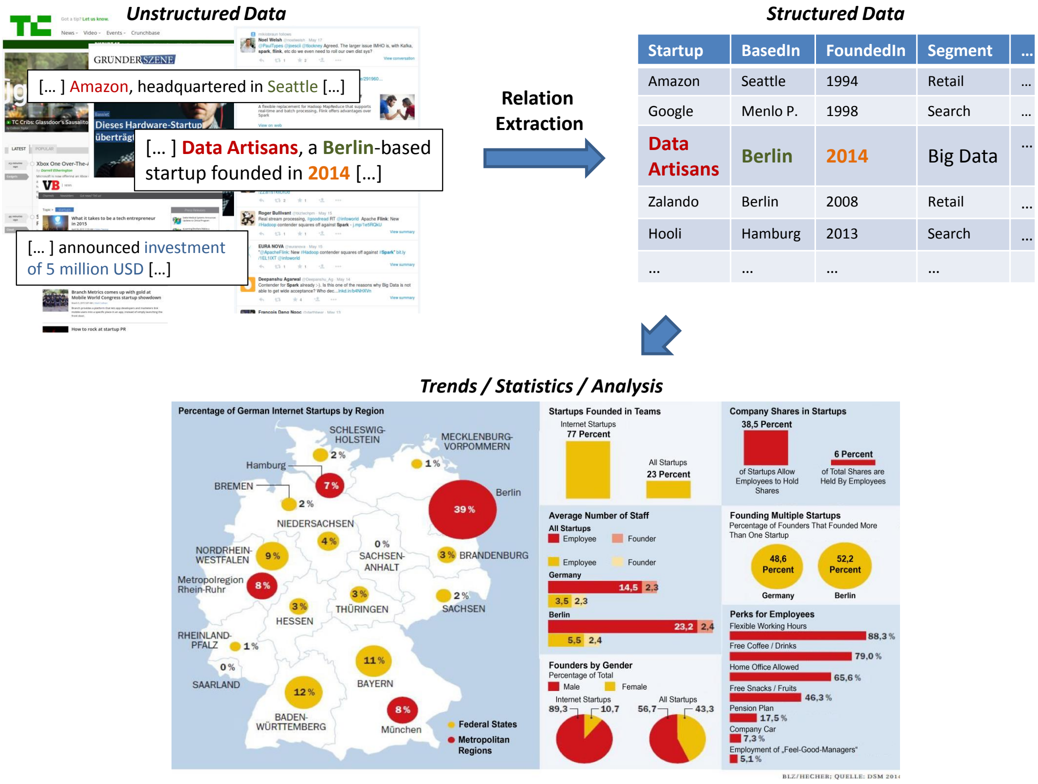
The world is awash in text. As the planet's population increasingly become interconnected through ubiquitous access to phone and internet, natural language remains the primary exchange format for information between humans. Each day, untold amounts of text data are user generated be it on Web pages, online news, forums, blogs, tweets, emails, text messages and chat rooms. For instance, [3] gather that *each minute* in 2014, users sent over 200 million emails and nearly 300,000 tweets, created over 25,000 reviews on YELP, and typed over 4 million search queries into GOOGLE. For 2014, the popular WORDPRESS blogging CMS reported that nearly 50,000 new blogs were created and over 1.5 million blog posts written every day [4]. As of writing, approximately 600 edits are made to WIKIMEDIA projects (which include Wikipedia) each minute of every day [5]. While only such fragmented estimates of the amounts of user generated text content on the Web and elsewere are available, its growth is estimated to accelerate rapidly [6].

Seen from a data mining perspective, such abundance of text data offers opportunities for analytics applications: Somewhere in this sea of data lie pieces of information – encoded in natural language – that may be useful to a particular information need.

For example, assume that a company specializes in investments in technology startups. In order to make informed investment decisions, such a company would like to draw up a range of statistics such as all recent startups (up to one year old) grouped by region and grouped by technology segment. However, much manual research is necessary to compile such a list: The company will need to find out what startups (worldwide) have been founded in the past year, in what cities they have been founded and what their technology segment is. Even more information might be useful to this hypothetical company: Information on the founders and their background, investors that either have invested or are planning to invest in a particular startup, growth and revenue numbers posted for each startup and so forth. The more distinct types of information are available, and the more comprehensive and up-to-date the available information is, the greater the potential for our hypothetical company to do analytics that support its investment decisions. However, in a fast-changing world in which innumerable startups appear and disappear, manually keeping track of all developments is not feasible.

This example points to the underlying challenge in text data analytics. On the one hand, we may assume that much of the information required for the use case above is reported in publicly available data such as newswire text, blogs and tweets. On the other hand though, this information is available only in unstructured form and therefore accessible to keyword search only, in the way of Web search engines. As such, this information cannot be placed into a structured database and is therefore inaccessible to querying, grouping, sorting, filtering, organizing and analyzing information in the ways required for the above use case.

Relation Extraction The task of Relation Extraction (RE) addresses this problem by creating *extractors* that automatically find instances of semantic relations in unstructured data such as natural language text [7]. Relations typically hold between two (or more) entities. An example extraction task is to find instances of the COMPANYBASEDINLOCATION relation, which relates a company to the location in which it is based. An extractor for this relation takes as input a text corpus and outputs any instances of the relation that it identifies. For example, it might find the two instances <*Starbucks, Seattle*> and <*Amazon, Seattle*>, indicating that both Starbucks and Amazon are based in the city of Seattle. Crucially, this information may be expressed in text in many different ways: The text fragments "***Seattle**-based **Starbucks***" and "***Starbucks** is headquartered in **Seattle***" both indicate the presence of a COMPANYBASEDINLOCATION relation instance, albeit in widely different wordings and syntax. An extractor must therefore be able to identify all possible ways for expressing a relation in natural language in order to effectively find relation instances. Relation instances extracted from text are *structured information*



Unstructured Data
[...] Amazon, headquartered in Seattle [...]
[...] Data Artisans, a Berlin-based startup founded in 2014 [...]
[...] announced investment of 5 million USD [...]
Relation Extraction
Structured Data
Trends / Statistics / Analysis
Percentage of German Internet Startups by Region
SCHLESWIG-HOLSTEIN 2 %
MECKLENBURG-VORPOMMERN 1 %
Hamburg 7 %
BREMEN 2 %
Berlin 39 %
NIEDERSACHSEN 4 %
SACHSEN-ANHALT 0 %
BRANDENBURG 3 %
NORDRHEIN-WESTFALEN 9 %
Metropolregion Rhein-Ruhr 8 %
THÜRINGEN 3 %
HESSEN 3 %
SACHSEN 2 %
RHEINLAND-PFALZ 1 %
SAARLAND 0 %
BAYERN 11 %
BADEN-WÜRTTEMBERG 12 %
München 8 %
Federal States
Metropolitan Regions
Startups Founded in Teams
Internet Startups 77 Percent
All Startups 23 Percent
Company Shares in Startups
38,5 Percent of Startups Allow Employees to Hold Shares
6 Percent of Total Shares are Held By Employees
Average Number of Staff
All Startups
Employee
Founder
Employee
Founder
Germany
14,5 2,3
3,5 2,3
Berlin
23,2 2,4
5,5 2,4
Founding Multiple Startups
Percentage of Founders That Founded More Than One Startup
48,6 Percent Germany
52,2 Percent Berlin
Founders by Gender
Percentage of Total
Male
Female
Internet Startups 89,3 10,7
All Startups 56,7 43,3
Perks for Employees
Flexible Working Hours 88,3 %
Free Coffee / Drinks 79,0 %
Home Office Allowed 65,6 %
Free Snacks / Fruits 46,3 %
Pension Plan 17,5 %
Company Car 7,3 %
Employment of „Feel-Good-Managers“ 5,1 %
BLZ/HECHER; QUELLE: DSM 2014

Startup	BasedIn	FoundedIn	Segment	...
Amazon	Seattle	1994	Retail	...
Google	Menlo P.	1998	Search	...
Data Artisans	Berlin	2014	Big Data	...
Zalando	Berlin	2008	Retail	...
Hooli	Hamburg	2013	Search	...
...	...	...	...	

FIGURE 1.1: Example of Relation Extraction from multilingual Web text. The extracted structured data is used for generating statistics and measuring trends (trends figure based on [8]).

and may therefore be directly input into a database. See Figure 1.1 for an example overview of extracting structured information from Web text and generating analytics.

Generally speaking, there are two principal methods for creating relation extractors that are established in scientific literature [9]:

In **machine-learning based approaches**, extractors are typically classifiers trained over labeled data in which they observe a set of lexical and syntactic features. The crucial bottleneck here is that labeled training data must first be produced at sufficient quality and quantity for every relation and domain of interest, a process that may be highly effort-intensive. Accordingly, much research focuses on ways of more inexpensively producing training data, for example through crowdsourcing [10], bootstrapping [11] or distant supervision [12], or minimizing the dependence on training data when adapting extractors from one domain to another [13].

In **rule-based approaches** on the other hand, humans manually build a rule-set of extraction patterns over lexico-syntactic feature sets [14]. While this approach requires no labeled data, it has well-known difficulties of scalability when rule and feature sets become too large and complex to be effectively managed. Current research on this line of approaches is scant (as most research is focused on machine learning based approaches), but recent developments focus on tooling and index structures to support the rule-writing process [15–17].

Limitations Both lines of approaches, however suffer from a number of limitations especially with regards to practical scenarios. The overarching problem is one of **cost**; the process of creating extractors requires either labeled data to be produced at sufficient quality and quantity in order to train a supervised machine learning algorithm [12, 18], or the manual creation of a complex set of extraction rules [14, 19]. In either case, the process is tedious and time-consuming and requires trained specialists with an extensive background in NLP, rule-writing or machine learning [9]. This expensive process needs to be repeated for every relation of interest, and every *language* of interest. For instance, for the same relation, separate extractors need to be created for English, German and Chinese text, further increasing the costs if text data is available in multiple languages [20].

Because of the high costs involved in creating extractors, great care must be taken when deciding which relation types to look for in a given corpus of text data. Practical scenarios are however often characterized by imprecise and rapidly changing information needs and uncertainty regarding the type of information contained in large, given text corpora [9]. Next to the issue of cost, this raises a number of difficulties when applying RE to practical scenarios:

Corpora of unknown content On of the major problems is how to reconcile imprecise information needs with corpora that, due to their very large size, are largely of unknown content. A typical first step in RE is to carefully define extraction tasks for a given information need. Only when the relations of interest are precisely defined, the creation of extractors can commence since it entails either producing large amounts of labeled training data or manually constructing complex extraction rule-sets. This costly two-step approach (first define relations, then build extractors) becomes impractical when large corpora of unknown content are involved since it is *a priori* unclear what types of structured information they contain. In the worst case this can mean that extractors are built at high cost only to discover that there are few instances of the pre-defined relation in the corpus. At the same time, this can mean that there are *other* relations in the corpus that would be valuable for a given information need that go unnoticed in the planning phase and are therefore not extracted. As I argue in this thesis, classical RE methods are too rigid to be used to work with corpora of unknown content, especially in the light of vague and shifting information needs.

Corpora of many languages Text data is increasingly available not only in ever larger quantities, but also in many different natural languages. Indeed, the example sources for text data listed in the opening lines of the introduction, such as Web text, tweets and newswire text, are readily available in a multitude of languages. The CLUEWEB09 reference corpus of Web data for instance contains text from 10 different major languages [21], the EUROPARL corpus of European parliamentary proceedings contains text in 26 distinct European languages [22], and as of writing there are over 270 active language editions of Wikipedia online [23]. While the distribution of languages on the Web can only be estimated, trends point towards a relative decline of English and a rise in use of non-English languages [24]. Revisiting the example use case of the company interested in global startups, we can see how multilingual data may be especially interesting; Chinese media may report on startup activities that English-language media do not, as do German, French, Russian and other national media. The challenge here is how to define extractors in such a way as to be able to handle multilingual data without incurring a blowup in costs.

1.2 Main Contributions

With this thesis, I report my research on a set of methods for Relation Extraction that address the above stated practical limitations, namely the high costs and the inability to handle unknown, multilingual corpora. The central challenge is to drive down costs for prototyping and developing extractors to such a degree that the above lamented two-step approach of first defining extraction tasks and then creating extractors can be replaced by a more natural way of working with unknown data. This means that I place the following *desiderata* on the researched approaches:

- Firstly, RE should not be limited to a pre-defined set of relations, but rather enable data scientists to openly *explore* the relational information contained in a large corpus of text data and *discover* new types of semantic relations that may meet their information need. For discovered relations, data scientist should be able to rapidly prototype extractors that run on the corpus and give an indication of the amount of relation instances the corpus contains.

- Secondly, this process should not require significant effort on part of the data scientist. In particular, this means that the data scientist should not be required to produce large amounts of labeled training data to train a classifier, nor should she need to manually construct rule-based extractors from scratch. In addition, the data scientist should not require an extensive background in machine learning or computational linguistics in order to explore and extract information from natural language data. Rather, the required effort should be reduced so that this type of data science becomes available to broader groups of non-specialists.

- Finally, Relation Extraction should not be limited to only one language, but rather automatically work for text in different human languages without additional effort – or knowledge of many languages – on the side of the data scientist. Indeed, the complexity of multilingual data should be hidden from the data scientist as much as possible.

In order to address these desiderata, I propose a process of *exploration* for relations of interest. Users work with unknown corpora without pre-specifying extraction tasks, but rather progressively discover relations of interest with a minimal number of interactions. A key challenge is to identify an abstraction layer for interactions that is both intuitive to use and powerful enough to handle linguistic diversity as well as *different languages*. A further key challenge is how to employ distributional evidence from available data in order to aid discovery and exploration of unknown corpora. In more detail, the main contributions of this thesis are:

1. Relation Discovery I investigate a fully unsupervised method for Relation Discovery (RD) in order to automatically identify prominent relations in an unknown corpus. The method is based on distributional evidence computed across the corpus and the use of clustering (i.e. unsupervised machine learning) to group correlating relation instances into discovered relations [25, 26] . In particular, I investigate the impact of an informed linguistic abstraction based on deep syntactic information and fine-grained entity type restrictions on Relation Discovery. An extensive experimental evaluation shows that the proposed approach outperforms earlier Relation Discovery efforts with less informed pattern generation approaches, and that the proposed abstraction layer is highly suited for Relation Extraction. I use the resulting state-of-the-art system for Relation Discovery to analyze strengths and limitations of fully unsupervised approaches.

This research has resulted in two full paper publications. In addition, I co-organized a conference workshop on this topic:

- Unsupervised Discovery of Relations and Discriminative Extraction Patterns. *Alan Akbik, Larysa Visengeriyeva, Priska Herger, Holmer Hemsen, Alexander Löser.* 24th International Conference on Computational Linguistics, COLING 2012.
- Effective Selectional Restrictions for Unsupervised Relation Extraction. *Alan Akbik, Larysa Visengeriyeva, Johannes Kirschnick, Alexander Löser.* 6th International Joint Conference on Natural Language Processing, IJCNLP 2013.
- Proceedings of the First AHA!-Workshop on Information Discovery in Text. *Alan Akbik and Larysa Visengeriyeva.* 25th International Conference on Computational Linguistics, COLING 2014.

2. Exploratory Relation Extraction The analysis of Relation Discovery strongly points to the need for limited user interactions in order to steer the discovery process towards a user-defined information need and to complement distributional evidence with domain knowledge. Based on this, I define the paradigm of *Exploratory Relation Extraction* (ERE), which incorporates elements from Relation Discovery, rule-based RE and pre-emptive Information Extraction [27] in order to maximize user influence while minimizing costs. I design a workflow for ERE that allows non-expert users to interactively explore a corpus with a minimal number of interactions, identify relations of interest and prototype extractors. I experimentally evaluate the approach on a very large dataset and find that the proposed approach effectively lowers the entry barriers for user-driven exploration and enables rapid prototyping of high-precision extractors.

This research has resulted in a full paper publication, as well as two demonstration papers:

- PROPMINER: A Workflow for Interactive Information Extraction and Exploration using Dependency Trees. *Alan Akbik, Oresti Konomi and Michail Melnikov.* The 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013.
- Exploratory Relation Extraction from Large Text Corpora. *Alan Akbik, Thilo Michael and Christoph Boden.* 25th International Conference on Computational Linguistics, COLING 2014.
- SCHNÄPPER: A Web Toolkit for Exploratory Relation Extraction. *Thilo Michael and Alan Akbik.* 53rd Annual Meeting of the Association for Computational Linguistics, ACL 2015.

3. Multilingual Semantic Role Labeling To address the issue of multilingual data, I propose a method for multilingual Semantic Role Labeling (SRL) that parses different languages into a *language-neutral* representation of shallow semantics [28]. This representation enables a shared feature space for different languages against which extractors can be developed. This would both enable us to hide language-specific elements from the data scientist and enable extractors to work across many different languages at no additional cost. However, while *language-specific* labeled data exists for training English SRL (as well as some other languages to a lesser degree) [29–33], no resources exist for training multilingual SRL systems. In order to enable the proposed parsing into a language-neutral representation, I propose an *annotation projection* approach that automatically creates the appropriate labeled training data from parallel corpora. I execute this approach for 7 distinct languages from different language groups, namely Arabic, Chinese, French, German, Hindi, Russian and Spanish in order to bootstrap semantic parsers for these languages. An extensive evaluation shows that the method outperforms earlier annotation project works and that the generated training data is of moderate to high quality, enabling the training of multilingual parsers for these languages.

This research has resulted in a full paper publication:

- Generating High Quality Proposition Banks for Multilingual Semantic Role Labeling. *Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan and Huaiyu Zhu.* 53rd Annual Meeting of the Association for Computational Linguistics, ACL 2015.

Together, the three contributions address the desiderata outlined above and allow for the exploratory analysis of large and multilingual datasets at low costs.

1.3 Thesis Structure

This thesis is structured in the following way: In Chapter 2, I begin with an overview over Relation Extraction, outline common lines of approaches and different types of features used to create RE systems and discuss limitations of existing approaches. I conclude the introduction with a summary of principal terminology and notation as used in this thesis.

I then present the three main contributions in the order presented above. In Chapter 3, I present my research in unsupervised methods for *Relation Discovery* [34–36]. Building on the results of this research, I present and evaluate the proposed paradigm of *Exploratory Relation Extraction* [37–39] in Chapter 4. I then present my research in *Multilingual Semantic Parsing* [40] in Chapter 5. All three contribution chapters follow the same structure: They each begin with an overview section in which a problem statement is presented, specific related work discussed and the contributions are listed. This is followed by a detailed method section and concluded with an evaluation section in which the experimental setup, the experiments and the results are presented and discussed.

The thesis concludes in Chapter 6 with a summary and an outlook into future directions of research.

2

Preliminaries

"Again the whirlwind!" the man cried inexplicably.
He's mad.
"All of this!" he ranted. "Every word a whip!"

– Excerpt from [2]

2.1 Overview

In this chapter, I give an overview of central concepts of this thesis. As the first major topic I introduce the task of Relation Extraction (Section 2.2): I begin with the task's early history by going through the Message Understanding Conferences from the 1980's and 1990's in Section 2.2.1. I then introduce standardized measures used to evaluate RE systems in Section 2.2.2. I conclude this topic with an overview of rule-based and machine learning-based RE approaches and a discussion on their advantages and disadvantages (Section 2.2.3).

As the second major topic I go through different types of Natural Language Processing (NLP) components from the point of view of building relation extractors (Section 2.3). I cover different levels of linguistic analysis, including lexical, syntactic and shallow semantic features. In particular, I introduce the dependency formalism as the main deep syntactic abstraction used in this thesis (Section 2.3.2). Building on this, I give an overview of Semantic Role Labeling (Section 2.3.3), a shallow semantic abstraction that is more language-invariant than syntactic parsing. This abstraction forms the basis of the multilingual research in this thesis.

Finally, I give a summary over key terminology and notation as used in this thesis.

2.2 Relation Extraction

Relation Extraction is the task of finding *instances* of a set of pre-defined *relations* from text. A *relation* is a semantic relationship type that holds between two or more entities. A *relation instance* (or *instance* for short) is an ordered set of entities for which a relation holds. For example, the BORNIN relation describes the relationship between a person and their place of birth. The *entity pair <Albert Einstein, Ulm>* is an instance of this relation, indicating that the person Albert Einstein was born in the city of Ulm [41–44].

Most recent work in RE focuses on such *binary* relations, i.e. relations that hold between two entities. Relations that hold between more than two entities are referred to as *N-ary* [45–47], an example of which might be a RELOCATED relation that indicates which person moved their residence from which old location to which new location. Early definitions of extraction tasks were based on filling so-called "templates" that had many "slots" [48–50] and thus could be considered N-ary relations. However, this was gradually abandoned to favor binary relations which in turn could be incorporated as elements into more complex templates [51].

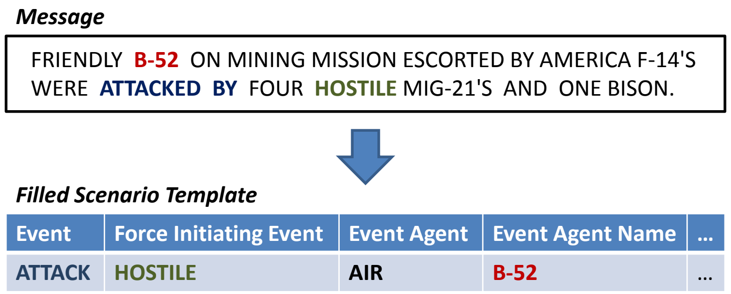
In the next section, I go through the early history of Information Extraction (IE), from which the subtask of RE was developed.

2.2.1 Early History

The task of Relation Extraction, as well as the supertask of Information Extraction, goes back to the Message Understanding Conferences (MUCs) of the late 1980s that were instituted by DARPA in response to the opportunities presented by the enormous quantities of on-line texts. It has been observed that through the focus on this type of task, DARPA created this field of study [50]. In the MUCs, the task was to create systems that find certain types of structured information in text.

2.2.1.1 MUC-1 and MUC-2: First Steps

The first two installments of the conference, held in 1987 and 1989 respectively, were of exploratory nature to establish the task and evaluation measures [50]. The conference was spearheaded by the Naval Ocean System Center (NOSC) of the US Navy, and supported by DARPA (the US Defense Advances Research Projects Agency). The task was created for two reasons: The first was to encourage research into methods that could extract valuable, structured information from increasing quantities of available texts, for example in military or news reports. The second was to encourage research into NLP technologies such as syntactic parsers, which were believed to be necessary in order to extract information from natural language data. By



Message

FRIENDLY B-52 ON MINING MISSION ESCORTED BY AMERICA F-14'S WERE ATTACKED BY FOUR HOSTILE MIG-21'S AND ONE BISON.

Filled Scenario Template

Event	Force Initiating Event	Event Agent	Event Agent Name	...
ATTACK	HOSTILE	AIR	B-52	...

FIGURE 2.1: Example scenario template filling task from MUC-2.

providing an end-to-end use case and establishing task-driven evaluation norms for comparatively evaluating different approaches, it was hoped that research into NLP components would be accelerated as well [48].

Task. The task was to fill *scenario templates* for a simple database update task [48]. Each scenario (or "event") represented one military "action" and had 10 slots to fill. Some of these were *set-fill slots* with a fixed enumeration of values, such as "Force Initiating Event" (either FRIENDLY or HOSTILE) or "Event Type" (either DETECT, TRACK, TARGET, HARASS, ATTACK, or OTHER). Others were *string-fill slots*, such as "Event Agent Name" required entry of either the canonical form of a name, a taxonomic category or an entity ID value. In cases of missing information, the value "NO DATA" could be entered into most of the 10 slots. Each filled template was added to the event database as a record in which the slots were the fields.

Data set. As text data for the first two MUCs, a narrow domain with short, simple messages was chosen, namely narrative lines in short naval messages about encounters with hostile forces. This had the advantages that the data reported on a limited set of topics and contained little embellishing information or speculation, i.e. the messages were short and succinct, with a fairly small vocabulary (3000 words). However, the "telegraphic" style of the messages also meant that there was little punctuation, heavy use of ellipsis (omission of syntactic constituents) and full-text capitalization [48, 49, 52]. A dataset of 125 of such Navy messages was provided as *development set* (distributed 3 months prior to the evaluation) and an additional 5 messages as *test set* used for on-site evaluation at the conference [53].

Figure 2.1 gives a simplified example of filling a scenario template from one Navy message

Evaluation. The first two conferences established initial measurements for evaluation of extraction quality, such as *precision* and *recall* (introduced in detail in Section 2.2.2). In addition, a number of other measurements were made to estimate the *generality* of the approach (by comparing recall and precision between development and test set), the *robustness* (by comparing recall and precision between original narratives as input and a manually cleaned up version of the narratives) and *progress* (measuring precision and recall at different stages of development).

2.2.1.2 MUC-3 and MUC-4: Evaluation Measures

The next two installments of the conference, held in 1991 and 1993 respectively, increased the complexity of the task by using a significantly larger dataset with a less narrow focus and accordingly a far broader vocabulary [52].

Task. The task was to find terrorism events in the news reports and fill templates for each event. In MUC-3, the event template contained 18 slots, such as “Type of incident” (a set-fill slot that could be one of BOMBING, KIDNAPPING etc.), “Perpetrator” (a string-fill slot) and “Location of incident” [49]. In MUC-4, the task complexity increased further to 24 slots. One reason for this increase was that MUC-3 slots containing composite values were split up. An example is the MUC-3 slot “Type of incident” filled with the value ATTEMPTED BOMBING, which was split into two MUC-4 slots: “Incident: Type” with the value BOMBING, and “Incident: Stage of execution” with the value ATTEMPTED. In addition, a few extra slots were defined for MUC-4, such as the string-fill slot “Incident: Instrument” for the instrument of attack [54].

Data set. For MUC-3, a corpus of 1,300 newswire reports of terrorist events in Central and South America was used as development set, with a vocabulary of over 18,000 words, i.e. one order of magnitute more than in MUC-2 [52]. The data source was a database of worldwide news gathered by the Foreign Broadcast Information Service of the U.S. Government [49]. A subset of newswire reports were retrieved from this database with a set of keyword queries which were combinations of target country names and words that indicate terrorism activities. The test set used for evaluation consisted of 100 additional news reports. For MUC-4, a second test set of 100 news reports from a different year was added in order to test generality of the approach [54].

Evaluation. In MUC-4 another crucial evaluation metric was introduced as a single-score measure to make the results of different systems directly comparable: The f-measure, which is the harmonic mean between precision and recall.

2.2.1.3 MUC-5: Multilingual Data

The next MUC was held in 1993 and focused more specifically on the difficulties of developing extractors for different domains and languages. As the dataset, again news reports were chosen, but this time of two different domains, namely reports on “business joint ventures” and “electronic circuit fabrication”. While previous MUCs had only looked at English-language data,

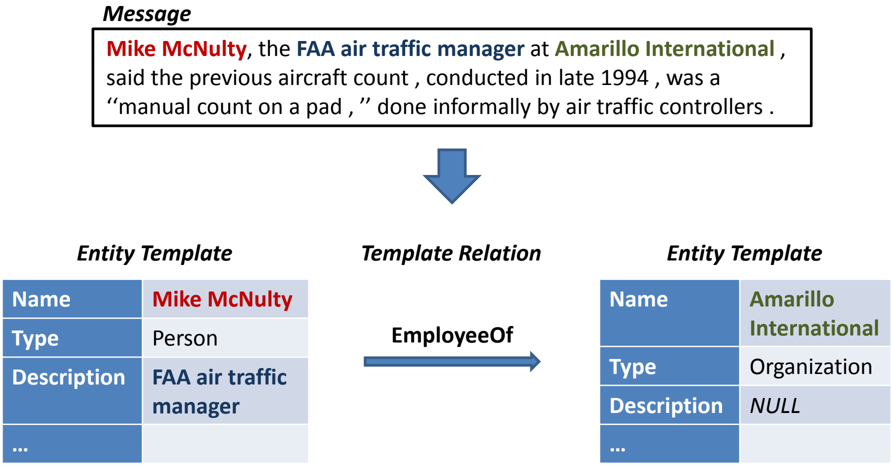

Message
Mike McNulty, the FAA air traffic manager at Amarillo International , said the previous aircraft count , conducted in late 1994 , was a "manual count on a pad , " done informally by air traffic controllers .
Entity Template
Name
Mike McNulty
Type
Person
Description
FAA air traffic manager
...
Template Relation
EmployeeOf
Entity Template
Name
Amarillo International
Type
Organization
Description
NULL
...

FIGURE 2.2: Example entity template relations from MUC-7 (message from [56]).

MUC-5 included data both in English and Japanese [50, 55], highlighting for the first time the need for multilingual IE.

Task. The goal was to increase the task realism with regards to the input data and the complexity of the extraction requirements [55]. The task now was to fill 11 distinct event templates with a total of 47 slots of different types: *Set-fill* slots that could be filled with one of a fixed set of values, *numeric/complex* slots to be filled with normalized values, *string-fill* slots to be filled with normalized strings such as corporation names and, for the first time, *pointer-fill* slots that pointed to other entries in the database. The latter allowed for a nested template structure and a more complex database.

2.2.1.4 MUC-6 and MUC-7: Information Extraction Subtasks

In MUC-5, most teams used relatively simple shallow pattern matching techniques which they adapted to the required complex template structure. While this approach had been noted to yield relatively good results, it was also effort intensive [51]: Teams spent more than 6 months in setting up their systems, some teams considerably more so. This raised the question of whether the effort involved was in fact prohibitive to many tasks. In order to reduce the overhead for setting up IE systems to specific templates and encourage the development of components that address specific NLP tasks, MUC-6 split the task of filling scenario templates into several *subtasks*, distinguishing between entity-level and scenario-level templates. MUC-7 further separated out the task of extracting relations between entities from scenario-level templates (which could include several relations), defining for the first time the task of **Relation Extraction** that is the focus of this thesis.

Tasks. In MUC-6, the first subtask was to recognize entity names, such as organizations, locations or persons, as well as temporal expressions and certain types of numerical expressions in text, a task now commonly referred to as *Named Entity Recognition* (NER). The second subtask was to fill entity-level templates, i.e. to identify entities and a set of attributes depending on the entity type (ORGANIZATION, PERSON or ARTIFACT) - For instance, an entity of type ORGANIZATION had attributes such as the company name, its alias and its location [57]. Entity-level templates were separated out from scenario-level templates in order to avoid redundancy (since the same entities might be involved in different scenarios). See Figure 2.2 for an example.

The third subtask in MUC-6 was to fill scenario templates that mediated a set of entity-level templates. In MUC-7, this task was further divided into two distinct tasks, one for *relations* and one for *events*. The Relation Extraction task was to find well defined relations between pairs of entities, such as EMPLOYEEOF, PRODUCTOF and LOCATIONOF. The event task was to fill event templates with entities and relations and most closely resembled the scenario template task of previous MUCs.

Data set. In MUC-6, the data set consisted of 318 annotated Wall Street Journal articles from the years 1993 and 1994, distributed by the Linguistic Data Consortium[1]. In order to test the portability of IE systems, the evaluation scenario was distributed only four weeks before the evaluation. It concerned changes in corpora executive management personnel [58]. In MUC-7, the scenario concerned reports on air vehicle launches and accidents. The data set consisted of approximately 158,00 articles retrieved from the New York Times News Service corpus using domain relevant terms. Two sets of training and test data were made available to train and evaluate the tasks.

2.2.1.5 ACE: Refinement of Relation Extraction Task

The subtasks were further explored and refined in the ACE (Automatic Content Extraction) program, a continuation of MUC that began with a pilot study in 1999 [59]. Notably, while each of the ACE tasks could be seen as continuations of the entity, relation and event-level extraction tasks respectively, the "entity template" task was not continued. Rather, the detection of entity-attribute relations in the entity templates became part of Relation Extraction. In addition, the scenario template task was simplified to favor event structures that more resembled N-ary relations than the complex template structure of the MUCs.

Task. In the initial period from 2000-2001 (ACE-02), the ACE effort was focused on the Entity Detection and Tracking (EDT) task, a continuation of the NER task from the MUCs in which mentions of entities of types PERSON, ORGANIZATION, GEOPOLITICAL, LOCATION, FACILITY or their subtypes needed to be recognized. In the following two years (ACE-03), the task

[1] https://catalog.ldc.upenn.edu/LDC2003T13, accessed 05/05/2015

	Named Entity	Template Element	Relation	Scenario / Event	Multilingual
MUC-3				yes	
MUC-4				yes	
MUC-5				yes	yes
MUC-6	yes	yes		yes	
MUC-7	yes	yes	yes	yes	
ACE-1	yes				
ACE-2	yes		yes		yes
ACE-3	yes		yes		yes
ACE-4	yes		yes	yes	yes
ACE-5	yes		yes	yes	yes
ACE-7	yes		yes	yes	yes
ACE-8	yes		yes	yes	yes

TABLE 2.1: Overview over MUC and ACE tasks [59, 61].

of Relation Detection and Characterization (RDC) relations were explored and added. The goal was to detect mentions of one of five general relations and their subtypes among entities. Starting in (ACE-04), the task of Event Detection and Characterization (EDC) was added in which the goal was to find mentions of events in which a number of entities participate, similar to the MUC scenario template task, but with a simpler event structure. These three main tasks (entities, relations and events) were continued to be evaluated until the last installment of ACE (ACE-08) in 2006 [60].

Dataset. The ACE differed to MUC in that broader domains of source text were considered, including newspaper texts, telephone speech transcripts, blogs and OCR outputs [59]. In addition, over the course of the 8 ACE installments, source text of different languages was used next to English, including Chinese, Arabic and Spanish. ACE-08 featured a dataset of over 10.000 documents each for English and Arabic [60].

2.2.1.6 Summary

The history of the MUCs and ACE installments, summarized in Table 2.1, illustrates how the task of Information Extraction was gradually refined and decomposed from template-filling to entity, relation and event-level subtasks. Since each subtask has different challenges, this separation of tasks allowed for more portable technologies to be developed. In particular, the task of Relation Extraction (RE) was developed out of the realization that after the recognition of entities in text, the next most important step was to develop technologies search for entity-entity or entity-attribute relations. These could then be used to fill entity and scenario-level templates.

	Correct	Incorrect
Retrieved	true positives (TP)	false positives (FP)
Not retrieved	false negatives (FN)	true negatives (TN)

TABLE 2.2: Contingency table for evaluating Relation Extraction.

In addition to defining the three subtasks, early work recognized the need for developing technologies that work on multiple languages: In MUC-5, Japanese text was considered as input, while many of the later ACEs worked with text in Chinese and Arabic, as well as Spanish. Next to the task definitions, the main evaluation measures of precision, recall and f-measure were established in the MUCs, which I briefly introduce in the next section.

2.2.2 Evaluation Measures

Relation Extraction systems are evaluated in terms of precision, recall and f-measure against a *gold standard dataset* (also referred to as *answer key* or *ground truth*). Such gold data contains annotations of all instances for each relation that an extractor is expected to find in the corpus. This is used to check whether the instances retrieved by an extractor are correct or incorrect. *True positives* (TP) are all retrieved instances that are part of the gold annotation and therefore correct. All other retrieved instances are *false positives* (FP). See the contingency table in Table 2.2 for an overview.

The *precision* is the portion of true positives among all found instances, i.e. the overall correctness of the approach. This is calculated as

$$\text{precision} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FP}} \tag{2.1}$$

The *recall* is the potion of true positives among all positives in the gold data, i.e. the overall completeness of the approach. This is calculated as

$$\text{recall} = \frac{\#\text{TP}}{\#\text{TP} + \#\text{FN}} \tag{2.2}$$

There exists a tradeoff between both these measurements. Systems that are optimized for high recall tend to casts the nets relatively wide, finding many relation instances at the cost of precision. On the other hand, systems optimized for high precision use restrictive patterns that tend to find fewer instances and result in lower recall. In order to have a one-number metric for comparing different approaches, the f-measure (also referred to as *f-score* or *F1*) was introduced, which is the harmonic mean between precision and recall.

$$\text{f-measure} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2.3}$$

However, the f-measure has been criticized in that it equally weighs precision and recall, which may not be appropriate for all cases since depending on the application, either high recall or high precision might be more important [11]. For instance, a Question-Answering system might place more importance on high precision since a user might perceive incorrect answers less well than missing answers. Similarly, as discussed in Chapter 3, a Relation Discovery effort may favor the discovery of high precision (and therefore clearly defined) relations, as opposed to more diluted, high recall relations. Another example of this is discussed in Chapter 5, in which (at least in intermediary steps of the proposed algorithm) high precision is more important than high recall. Nevertheless, the f-measure is the community-accepted single-score measure for comparing RE systems [12, 62–64]. I therefore use all three metrics throughout this thesis.

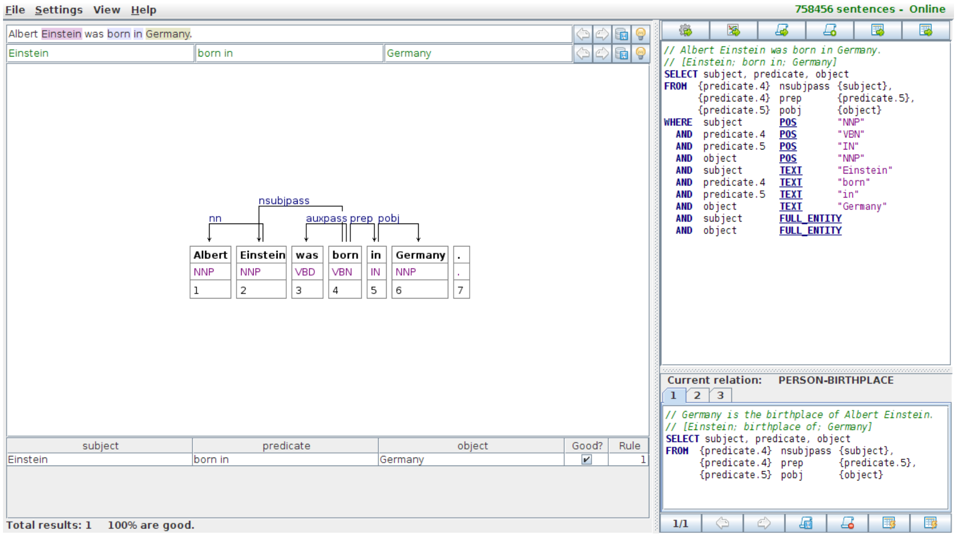
2.2.3 Methods

Relation Extraction systems take as input unstructured or semi-structured text data in which entities have already been recognized. For each co-occurring pair of entities, the challenge is to determine whether one of a set of pre-defined relations holds.

Assume for example an input text with two entities: "***Albert Einstein****'s birthplace* ***Ulm***". Assume furthermore that we are looking for instances of two relations: BORNIN and DIEDIN. A RE system must make the decision of whether the entity pair <*Albert Einstein, Ulm*> observed in this text fragment is an instance of one, both or none of these relations. This decision is typically based on a set of lexico-syntactic features observed with an entity pair. For instance, in the above sentence, the word "*birthplace*" is observed between the two entities, a lexical feature that indicates the presence of the BORNIN relation.

The challenge of building Relation Extraction systems are twofold: On the one hand, a good set of lexical, syntactic or even shallow semantic features must be identified that reliably indicate the presence of relation instance[2] [18, 65, 66]. On the other hand, a mechanism must be found that makes the decision based on this feature set of whether an entity pair belongs to a certain relation. For this, there are two broad lines of RE approaches: *Rule-based* systems in which this decision is manually encoded through a set of rules. And *machine learning-based* approaches in which a classifier is trained on labeled training data that is either manually or semi-automatically produced. In the following, I go through archetypical Relation Extraction approaches and highlight strengths and weaknesses of each approach.

[2] A discussion of different classes of features follows in Section 2.3



Albert Einstein was born in Germany.

Einstein | born in | Germany

```
// Albert Einstein was born in Germany.
// [Einstein; born in; Germany]
SELECT subject, predicate, object
FROM  {predicate.4}  nsubjpass {subject},
      {predicate.4}  prep      {predicate.5},
      {predicate.5}  pobj      {object}
WHERE  subject       POS       "NNP"
  AND  predicate.4   POS       "VBN"
  AND  predicate.5   POS       "IN"
  AND  object        POS       "NNP"
  AND  subject       TEXT      "Einstein"
  AND  predicate.4   TEXT      "born"
  AND  predicate.5   TEXT      "in"
  AND  object        TEXT      "Germany"
  AND  subject       FULL_ENTITY
  AND  object        FULL_ENTITY
```

subject	predicate	object	Good?	Rule
Einstein	born in	Germany	☑	1

Total results: 1   100% are good.

```
// Germany is the birthplace of Albert Einstein.
// [Einstein; birthplace of; Germany]
SELECT subject, predicate, object
FROM  {predicate.4}  nsubjpass {subject},
      {predicate.4}  prep      {predicate.5},
      {predicate.5}  pobj      {object}
```

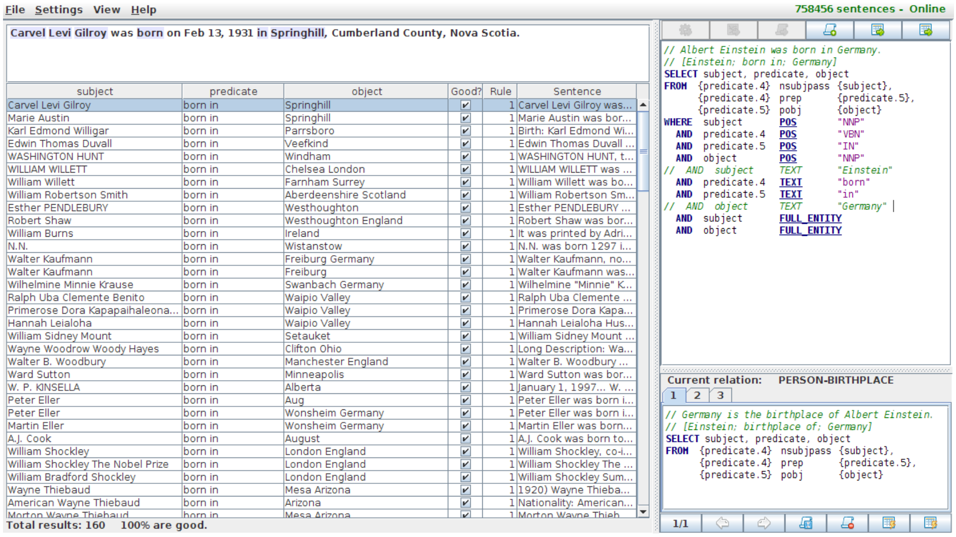
FIGURE 2.3: Main view of PROPMINER. The first steps of the workflow are executed here. Users enter a sentence in the top input field and annotate the subject, predicate and object for the desired relation. A rule is generated and displayed in the upper right panel. The lower right panel is the repository of already created rules. The parse of the input sentence is displayed in the center panel.

2.2.3.1 Rule-Based

In rule-based RE, humans manually compile a set of rules using lexico-syntactic features that determine whether an entity pair is an instance of a relation [14, 67]. Since rule-sets are often based on pattern-matching, rules are also thought of as extraction *patterns*. Depending on the RE system, matching rules are executed sequentially or even in a *cascading* fashion in which rules are embedded within other rules; for instance, a rule-set that matches BORNIN relation instance might embed a rule-set that finds PERSON entities [67, 68].

Typically, rule-based RE systems must strike a balance between the *expressive power* they offer in creating rule-sets and their *manageability*. If rule-sets become too complex, involving for instance too many features and embedded rules, they become less humanly readable, and are therefore more costly to manage, debug or extend [9]. Because of this, rule-based RE approaches have well-known difficulties of scalability [9]. Current research strives to address these problems by developing tooling and workflows to facilitate flexible incorporation of NLP components into the feature extraction step [14, 69], to assist the rule-writing process [15, 17, 39] and to inspect extractors for error-analysis [16].

Example rule-based RE system: PROPMINER I give an example of a rule-based RE system developed within this thesis as a preliminary study of using dependency tree features in patterns, called PROPMINER [39]. The central idea in this system is to enable an *example-driven workflow*



Carvel Levi Gilroy was born on Feb 13, 1931 in Springhill, Cumberland County, Nova Scotia.

subject	predicate	object	Good?	Rule	Sentence
Carvel Levi Gilroy	born in	Springhill	☑	1	Carvel Levi Gilroy was...
Marie Austin	born in	Springhill	☑	1	Marie Austin was bor...
Karl Edmond Willigar	born in	Parrsboro	☑	1	Birth: Karl Edmond Wi...
Edwin Thomas Duvall	born in	Veefkind	☑	1	Edwin Thomas Duvall ...
WASHINGTON HUNT	born in	Windham	☑	1	WASHINGTON HUNT, t...
WILLIAM WILLETT	born in	Chelsea London	☑	1	WILLIAM WILLETT was ...
William Willett	born in	Farnham Surrey	☑	1	William Willett was bo...
William Robertson Smith	born in	Aberdeenshire Scotland	☑	1	William Robertson Sm...
Esther PENDLEBURY	born in	Westhoughton	☑	1	Esther PENDLEBURY ...
Robert Shaw	born in	Westhoughton England	☑	1	Robert Shaw was bor...
William Burns	born in	Ireland	☑	1	It was printed by Adri...
N.N.	born in	Wistanstow	☑	1	N.N. was born 1297 i...
Walter Kaufmann	born in	Freiburg Germany	☑	1	Walter Kaufmann, no...
Walter Kaufmann	born in	Freiburg	☑	1	Walter Kaufmann was...
Wilhelmine Minnie Krause	born in	Swanbach Germany	☑	1	Wilhelmine "Minnie" K...
Ralph Uba Clemente Benito	born in	Waipio Valley	☑	1	Ralph Uba Clemente ...
Primerose Dora Kapapaihaleona...	born in	Waipio Valley	☑	1	Primerose Dora Kapa...
Hannah Leialoha	born in	Waipio Valley	☑	1	Hannah Leialoha Hus...
William Sidney Mount	born in	Setauket	☑	1	William Sidney Mount ...
Wayne Woodrow Woody Hayes	born in	Clifton Ohio	☑	1	Long Description: Wa...
Walter B. Woodbury	born in	Manchester England	☑	1	Walter B. Woodbury ...
Ward Sutton	born in	Minneapolis	☑	1	Ward Sutton was bor...
W. P. KINSELLA	born in	Alberta	☑	1	January 1, 1997... W. ...
Peter Eller	born in	Aug	☑	1	Peter Eller was born i...
Peter Eller	born in	Wonsheim Germany	☑	1	Peter Eller was born i...
Martin Eller	born in	Wonsheim Germany	☑	1	Martin Eller was born...
A.J. Cook	born in	August	☑	1	A.J. Cook was born to...
William Shockley	born in	London England	☑	1	William Shockley, co-i...
William Shockley The Nobel Prize	born in	London England	☑	1	William Shockley The ...
William Bradford Shockley	born in	London England	☑	1	William Shockley Sum...
Wayne Thiebaud	born in	Mesa Arizona	☑	1	1920) Wayne Thieba...
American Wayne Thiebaud	born in	Arizona	☑	1	Nationality: American...
Morton Wayne Thiebaud	born in	Mesa Arizona	☑	1	Morton Wayne Thieb...

Total results: 160   100% are good.

```
// Albert Einstein was born in Germany.
// [Einstein; born in; Germany]
SELECT subject, predicate, object
FROM  {predicate.4} nsubjpass {subject},
      {predicate.4} prep      {predicate.5},
      {predicate.5} pobj      {object}
WHERE  subject       POS       "NNP"
  AND  predicate.4   POS       "VBN"
  AND  predicate.5   POS       "IN"
  AND  object        POS       "NNP"
//  AND  subject     TEXT      "Einstein"
  AND  predicate.4   TEXT      "born"
  AND  predicate.5   TEXT      "in"
//  AND  object      TEXT      "Germany"
  AND  subject       FULL_ENTITY
  AND  object        FULL_ENTITY
```

Current relation: PERSON-BIRTHPLACE

1 2 3

```
// Germany is the birthplace of Albert Einstein.
// [Einstein; birthplace of; Germany]
SELECT subject, predicate, object
FROM  {predicate.4} nsubjpass {subject},
      {predicate.4} prep      {predicate.5},
      {predicate.5} pobj      {object}
```

FIGURE 2.4: Corpus view of PROPMINER, where extraction rules are modified and evaluated. The center panel is a table that shows the extraction results for the current rule. Users can inspect each extracted triple by clicking on the row. This displays the sentence in which the triple was found.

in which human-readable extraction patterns are pre-generated from annotated examples. Rule-writers can modify these rules and at each modification use them to query a large corpus for relation instances. When satisfied with the retrieved instances, users can save this pattern as a component in an extractor.

I illustrate this with an example for BORNIN as the target relation. Rule-writers begin with an archetypical sentence for the target relation, such as "*Albert Einstein was born in Germany*" and enter it into the main view of the tool, illustrated in Figure 2.3. They mark entities of interest in the sentence ("*Albert Einstein*" and "*Ulm*"), as well as all words that express the BORNIN relation in this sentence ("*born in*"). The tool generates an extraction rule from this input, as indicated in the upper right panel in Figure 2.3. The rule is *overspecified* with regards to all features: It requires the identical dependency subtree as observed in the example sentence, and the same POS tags and lexical values of all words involved in the tree. It will therefore only match sentences that are near-exact copies of the this sentence.

In a second step, users can now modify this rule to relax the matching conditions. For instance, users can comment out the conditions on the lexical values of the subject and the object. The rule then matches any sentence in which a similar dependency subtree as in the example sentence is observed, but with different entities. This is illustrated in Figure 2.4: The modified rule is in the upper right panel and the matching sentences are listed in the central panel. Users can browse matching sentences to determine if a rule correctly finds instances of the desired relation or whether further modifications are necessary.

This example illustrates how rule-based RE is driven by human inspection of data and domain-knowledge: In order to be able to write good extractors, humans need to know how a relation is expressed in text and what types of features are involved. Rule-based RE is therefore also considered to be *knowledge-driven*, as opposed to the *data-driven* techniques I introduce in the next section.

2.2.3.2 Machine Learning-Based

Machine Learning-based (*ML-based*) RE approaches were developed because of the high effort associated with manually writing extraction rule-sets [44]. In ML-based RE, classifiers are trained using a large set of labeled positive and negative training examples [70, 71]. These examples come from a *gold* dataset that takes the form of text in which all entities and relation labels are marked. The main challenges in ML-based RE are twofold: Firstly, for a relation of interest a good classification algorithm and feature-set must be identified, typically through a process of experimentation with gold data. Secondly, large amounts of good data need to be produced in order to train and evaluate extractors.

Especially the second issue is a crucial bottleneck to ML-based RE, since the costs of manually creating gold data tend to be very high [72]. In the following, I introduce a number of approaches to ML-based RE which address this problem in different ways.

Supervised Approaches The classical ML approach is fully supervised, meaning that classifiers are trained using a fully labeled dataset. This has the advantages that *cross-validation* can be used to automatically evaluate different feature sets and classification approaches [73]. In the typically used *10-fold cross validation*, the gold data is randomly divided into *training* and *test* sets. The training set is used to train the classifier, while the test set is used for automatic evaluation. This process is repeated ten times, which typically a ratio of 90% of the gold data used for training, and the remaining 10% used for testing. The average precision, recall and f-measure values are computed over all ten runs. This practice makes experiments reproducible if gold data is available.

However, a disadvantage are the high costs associated with manually labeling gold data if none exists. This is especially problematic since gold data needs to be created for every relation and *domain* of interest [71]. For instance, a WORKSFOR relation between a person and its employer might be differently expressed in the domain of newswire text than in the domain of forum posts or tweets. Worse, if multilingual data is considered, labeled data must additionally be created for every human *language* of interest. Because of this limitation, much research has focused on ways of more inexpensively producing training data, which I introduce in the following:

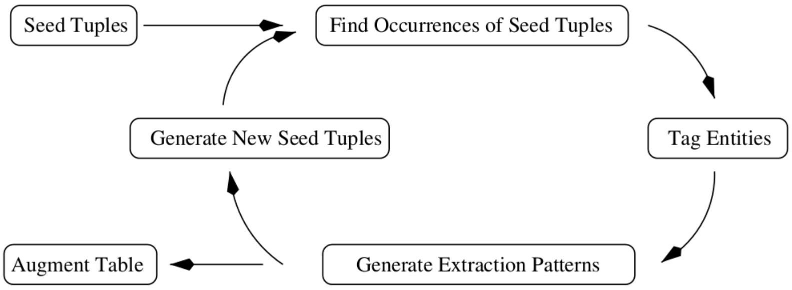


FIGURE 2.5: Schematic outline of bootstrapping approach as used in the SNOWBALL system (Figure taken from [42]).

Bootstrapping Bootstrapping approaches seek to minimize the effort for creating ML-based relation extractors [42, 43, 74–76]. Specifically, they seek to circumvent the need for gold data by using a small sample of relation instances, also referred to as *seed tuples* or *seeds*, to retrieve sentences that contain mentions of the relation instance. They then *tag entities* in these sentences, since relevant patterns can be learned from tagged sentences retrieved with relation seeds [74]. By applying these patterns to a corpus, additional instances of the relation of interest are mined, which are then added to the seed set. In the next iteration, the larger seed set is used to retrieve more sentences which are used to train even better classifiers which in turn find more seeds, and so on. Refer to Figure 2.5 for a schematic overview of the approach.

For instance, the entity pair <*Albert Einstein, Ulm*> can be a seed of the BORNIN relation. It can be used to retrieve sentences that express the relation of interest, such as "*Albert Einstein was born in Ulm*". A classifier might learn from a set of retrieved sentences that the lexical sequence "*born in*" is an important feature for the BORNIN relation. Applied to a corpus this pattern can be used to find additional instances from sentences such as "*Angela Merkel was born in Hamburg*", which are added to the seed set. The process is then repeated: A larger set of seeds retrieves more sentences from which a (presumably better) classifier is trained.

A known problem for such approaches is *semantic drift* [42, 43, 76]: At each iteration there is a risk that false positives will be added to the seed set. Such false positives will then be used to retrieve sentences that point to different relations, causing the classifier to be of lower quality and finding more seeds that belong to different relations. As this error propagates from iteration to iteration, this will cause the semantics of the extractor become diluted. This risk is especially high since bootstrapping mostly operates with positive training examples, i.e. there are no high quality negative training examples that can effectively counteract semantic drift. Strategies for containing this risk have been proposed that effectively reduce the search space, i.e. requiring long and overly specific patterns that match only a high quality subset of sentences [43], estimating pattern confidence using limited closed-world knowledge [77] or using a pre-computed clustering of patterns to guide the bootstrapping process [78]. However, they found that such methods only partially diminish the effect of semantic drift and come at a cost of recall [43, 77, 78].

Distant Supervision A related line of approaches uses existing knowledge bases (KBs) to provide supervision to learning extractors [12, 79–82]. Instead of manually providing a small set of seeds as in bootstrapping, a large set of seeds is retrieved by looking up instances of a relation of interest in a KB. These relation instances are used to retrieve large numbers of sentences which are automatically labeled as positive training examples. This data is used to train a classifier. Unlike bootstrapping, the initial set of seeds is large enough so that the process does not need to be repeated. For instance, by looking up the BornIn relation in KBs such as Yago [83] or Freebase [84], tens of thousands of relation instances can automatically be retrieved that serve as a very large seed set. Because of this, distant supervision is conceptually more resilient against semantic drift than bootstrapping approaches.

However, the underlying assumption that sentences retrieved with seeds express the correct relation is not always correct. For instance, we found the BornIn relation to correlate heavily with the DiedIn, LivesIn and IsGovernorOf relations [34], meaning that there are many persons who were born in, lived in, became governors of and died in the same place. Furthermore, a manual inspection of 200 sentences retrieved using distant supervision (discussed in more detail in Chapter 3.3.1.1) revealed that only approximately 76% pointed to the correct relation. This means that the positive training examples used for classification are in fact quite noisy. In addition, some distant supervision approaches create negative training examples from missing entries in KBs. However, since KBs are likely to be incomplete, missing data cannot reliably be used to generated negative training data, again potentially generating noisy data [82]. These problems cause the semantics of extractors trained using distant supervision to become diluted. In fact, it has been experimentally verified that precision decreases with the size of the set of retrieved sentences [10].

2.2.4 Summary and Discussion

The task of Relation Extraction was defined as a crucial subtask of Information Extraction through the course of the Message Understanding Conferences. The shift from scenario-level template filling to the subtasks of first recognizing Named Entities and then extracting relations between entities, allowed for the development of more portable technologies. It was recognized that technologies for RE could be used to find entity-entity and entity-attribute relationships which in turn could be used to populate more complex event-style templates.

$V \mid VP \mid VW^{*}P$
V = verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. marker)

FIGURE 2.6: Regular expression of POS tags used to validate between-text for RE. (Figure taken from [87]).

One of the main avenues of research in the past decades has been the question of how to create high quality extractors (measured in terms of precision and recall) without incurring the high costs of manually writing rule-sets or annotating labeled training data. A number of approaches to lower the effort required to create extractors, such as bootstrapping and distant supervision, have been proposed that trade off effort with extractor quality. However, none of these approaches provides a 'silver bullet', each having advantages and disadvantages of their own. Two limitations of these approaches stand out with regards to the goals of this thesis: (1) Relations of interest need to be pre-specified in advance. (2) Training data or rules must be generated at significant cost. As I argued in Chapter 1, this makes these approaches unsuitable for the task of open-ended *exploration*, which is necessary to work with vague information needs and unknown corpora.

There exists another line of approaches not covered in this chapter that does not require such pre-specification, called "Unsupervised Relation Extraction". It is the starting point of the research in this thesis and is therefore covered in detail in Chapter 3.

2.3 Lexical, Syntactic and Semantic Features

In the previous section, I introduced the task of Relation Extraction with regards to evaluation measures and general lines of approaches. As the second major topic in this chapter, I now take a look at different types of Natural Language Processing from the point of view of feature extraction for RE. I begin with shallow, word-level features and explain how such features can be used in extraction patterns. I then give a brief overview of deep syntactic analysis via dependency trees and explain how patterns can be defined that incorporate such information. Finally, I give an overview of Semantic Role Labeling (SRL), a shallow semantic abstraction layer that builds on predicate-argument structure. While I discuss each of these features in turn, it must be noted that current RE systems typically use a combination of different feature types [12, 26, 85, 86].

2.3.1 Shallow Features

Word-level, or "shallow", features have been used since the earliest work in Information Extraction [85, 86, 88]. An example is the use of *dictionaries* to identify proper nouns (sequences of capitalized words not in a dictionary) and so-called *trigger words*, e.g. words that indicate the presence of a relation [85, 88, 89]. For instance, the word "bomb" might indicate that a nearby proper noun should fill a slot in a bombing scenario template. Next to trigger words, a number of works in RE have identified the words between two entities as an important feature since they typically relate the two entities [25, 90, 91]. For instance, in the sentence "***Albert Einstein*** *was born in* ***Ulm***", a lexical pattern that holds for the entity pair *<Albert Einstein, Ulm>* is "*[X] was born in [Y]*", where [X] is the placeholder for the subject entity and [Y] is the placeholder for the object entity in the relation. Beyond the "between-text", other work in lexical patterns noted that important lexical information may be found before the first or after the second entity [26, 43]. For instance, [65] noted that there are three general types of lexical patterns:

Between : The first uses only the text between two entities as observed in text. For instance, the patterns "*[X] married [Y]*", "*[X] is married to [Y]*" or "*[X] is the husband of [Y]*" may indicate the presence of a MARRIEDTO relation.

Fore-Between Important information may also be found before the first entity in addition to the between-text. Examples of this are the patterns "*wedding of [X] and [Y]*" and "*marriage of [X] to [Y]*" again for the MARRIEDTO relation.

Between-After : Another type of pattern uses the text after the second entity in addition to the between-text. Examples of this are the patterns "*[X] , [Y] 's wife*", "*[X] whom [Y] married*" and "*[X] and [Y] 's wedding*".

However, early work also noted that purely lexical features or patterns suffer from recall issues since due to data sparsity many important lexical items are often not observed in the training data [63, 65]. To alleviate this, word-level analysis such as **stemming** or **lemmatization**[3] are used to find the base forms, or *stems/lemmas*, of words. For instance, the words "bombs", "bombed" and "bombing" may be interpreted with the same stem ("bomb"), potentially increasing the recall of dictionaries [88].

Next to lexical features, the most important shallow features are **Part-of-speech (POS)** tags. POS-tagging is a form of shallow syntactic analysis that determines the syntactic type of each word, such as verb, noun, adjective, proper noun and so on. Much like lexical features, shallow

[3] Lemmatization finds the linguistic base forms of words while stemming applies heuristics that chop off the ends of words.

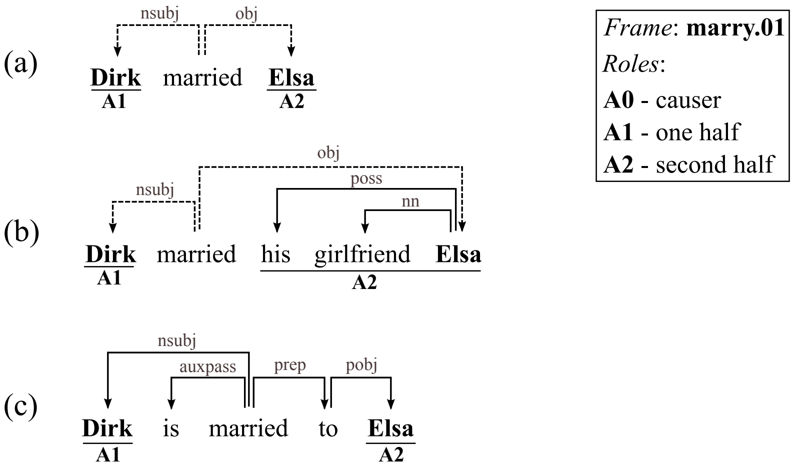


FIGURE 2.7: Three example sentences that express the MARRIEDTO relation for <*Dirk, Elsa*>. Dependency parse information is given above the sentences in arcs. Semantic Role labels are given below the sentences, with information on the frame "marry.01" in the upper right corner.

syntactic analysis has been used since early work in Information Extraction [88]. Such information can be used to identify so-called *chunks*, i.e. groups of nouns and verbs that together form a syntactic unit, which has been shown to be important if only shallow features are used [63]. Other work has observed that relations are often expressed using certain sequences of POS tags. [87] noted that between-text with certain sequences of POS tags reliably indicate the presence of a relation; For instance if the between-text consists of a verb followed by a preposition ("worked for", "traveled to", etc.), this often points to a relation between two entities. They created a regular expression of syntactic word types which they used to verify the between-text (illustrated in Figure 2.6) and observed significant increases in RE quality.

2.3.2 Dependency Trees

A limitation of shallow features is that long-range syntactic or semantic relationships between sentence constituents are not captured [18, 92]. This causes purely shallow patterns to be unable to handle a range of linguistic phenomena such as inserted clauses, appositions or adjectives.

Take for instance the first two sentences in Figure 2.7, which express a MARRIEDTO relation for the entity pair <*Dirk, Elsa*>. As established in the previous section, a simple between-text pattern for this relation is "*[X] married [Y]*". However, as can be seen in Figure 2.7, this pattern matches only one of the sentences, namely sentence *(a)*. Even through sentence *(b)* expresses the same relation, the inserted text "*his girlfriend*" prohibits a purely shallow pattern from matching.

One would need to define a second shallow pattern, namely "*[X] married his girlfriend [Y]*", to capture sentence *(b)*.

A more elegant solution to this problem is possible by identifying grammatical relations between words in a sentence. In Figure 2.7, the grammatical relations (also referred to as *dependencies*) are illustrated as arcs above the words. The arcs have labels that indicate the type of grammatical relationship. In sentences *(a)* and *(b)* for instance, the word "*Dirk*" is the grammatical subject (indicated by the label *nsubj*) of the verb "*married*". Likewise, the word "*Elsa*" is in both sentences the direct object of the verb. This means that it is possible to determine that sentences *(a)* and *(b)* have the same syntactic structure and that the apposition "his girlfriend" does not change the fact that "*Elsa*" is the grammatical object in both cases. This allows us to define a lexico-syntactic pattern in which we look for the grammatical subject and object of the verb "marry" as pattern for the MARRIEDTO relation, indicated by dotted arcs over sentences *(a)* and *(b)*. As the example illustrates, such patterns are more robust against linguistic variation.

Dependency parsing. One such analysis is dependency parsing, which determines the dependency structure of a sentence. Dependency grammar goes back to the seminal work of French linguist Lucien Tesnière who described syntactic structure as words connected by directed links of different types [93]. Each link connects a head word to one dependent (child). Each word may only have one head but any number of children, yielding a tree structure of dependency links in which verbs form the structural center, as well as deverbal nouns to a lesser degree. The labels of the dependency links indicate the type of syntactic relationship between two words.

The task of dependency parsing has been well researched in the NLP community [94, 95], yielding a large number of freely available dependency parsers. In this thesis, I conduct experiments with the CLEARNLP parser for English [96], as well as the STANFORD parser [97], the MALT parser [98] and the MATE family of parsers [99, 100] for other languages. I look into using dependency tree patterns in both unsupervised and rule-based approaches to RE and find that use of this abstraction greatly increases both RE quality and ease-of-use.

2.3.3 Semantic Role Labels

Semantic Role Labeling (SRL), also sometimes referred to as *shallow semantic parsing*, represents a further level of abstraction into parsing a sentence [101]. Instead of grammatical relations between words as in dependency parsing, SRL focuses on identifying the predicate-argument structure of sentences with semantic labels. The motivation for SRL is that shallow semantics and syntax correlate often, but not always, making a more semantics-oriented level of annotation necessary.

Frame: **open.01**

Roles:

A0 - *opener*

A1 - *thing opening*

A2 - *instrument*

A3 - *benefactive*

Examples:

$\underset{\textbf{A0}}{\underline{\text{The man}}}$ ***opened*** $\underset{\textbf{A1}}{\underline{\text{the door}}}$ with $\underset{\textbf{A2}}{\underline{\text{the key}}}$.

$\underset{\textbf{A1}}{\underline{\text{The door}}}$ ***opened***.

$\underset{\textbf{A2}}{\underline{\text{The key}}}$ ***opened*** $\underset{\textbf{A1}}{\underline{\text{the door}}}$.

FIGURE 2.8: Diathesis alternation correctly labeled using SRL.

Revisiting the three example sentences in Figure 2.7, we note that while the same dependency relation is found in sentences *(a)* and *(b)* (dotted arcs), sentence *(c)* is syntactically different: here, the main verb is a passive voice construction with a passive subject and a prepositional object. This syntactic difference is reflected in a different dependency tree. A dependency tree pattern that matches sentences *(a)* and *(b)* will therefore not match sentence *(c)* even though they are semantically similar.

In SRL, constituents are labeled with regards to their function (*role*) in semantic frames, which in turn are evoked by words in a sentence. In all three example sentences above, the frame-evoking element is the verb ”*marry*“. It evokes the frame MARRY.01, which represents the semantic notion of ”someone marries someone to someone“. Accordingly, the frame can bind arguments of three different roles to itself, labeled **A0** to **A2**. The frame is illustrated in the upper right corner of Figure 2.7 and the frame-semantic labels are given below the three sentences. As can be seen, this level of analysis is stable across all three sentences: Even through sentence *(c)* is in passive voice, the word ”Elsa“ is correctly identified as the **A2** (i.e. the ”second part“) of the ”marry“ frame. An extraction pattern defined over such shallow semantic labels will therefore be even more robust against linguistic variation and work for all three sentences in this example [102].

Diathesis alternation. A more complex example than active-passive switches are *diathesis alternations* in which the same grammatical function may evoke different roles even within the same voice [103]. The following sentences illustrate how semantics and syntax may deviate in such cases:

1. ”*The man opened the door with the key.*“
2. ”*The door opened.*“
3. ”*The key opened the door.*“

In all three cases, the semantics are similar in the sense that it is always a *door* that opens. However, the word ”*door*“ can be both the grammatical object (sentences 1 and 3) as well as the grammatical subject (sentence 2) of the verb "*open*". Similarly, the *key* that opens the door has the same semantic function in sentences 1 and 3, but different syntactic functions (prepositional object in sentence 1, subject in sentence 3). As illustrated in Figure 2.8, SRL finds the correct labels regardless of syntactic function and is therefore more invariant across linguistic variation.

SRL has been shown to be useful to a variety of NLP tasks such as Information Extraction and Question Answering [102, 104, 105]. In this thesis, I make use of two different SRL systems for English, namely CLEARNLP [96] and MATE-SRL [28] and investigate the creation of *multilingual* SRL as a step towards RE on multilingual datasets (Chapter 5).

2.3.3.1 Resources

There exist two major projects for determining the set of all possible English frames, namely PROPBANK [29] and FRAMENET [106]. Both are longstanding manual efforts with differences in philosophy [107].

FrameNet. In FRAMENET, frames are more abstract and global, meaning that the same frame may be evoked by different lexical units, including verbs, nouns and adjectives. Frames evoking elements were determined by inspecting usage examples of words that were believed to have semantic overlap and dividing them into subgroups of similar meaning. They were then inspected to identify frame elements that these frames could bind to themselves, distinguishing between three levels of importance for frame elements [106, 108]. For example, the verb "*open*" is grouped with verbs such as "*fasten*", "*button*", "*tie*" and "*uncork*" into evoking the *Closure* frame which binds frame elements such as *Agent* (the agent that opens or closes an item), *Fastener* (the fastener that the agent manipulates) and *Containing_object* (the item that is closed by the agent with the fastener). However, due to the abstract and global nature of the frames, FRAMENET has been shown to be impractical with regards to manual annotation, making it is difficult to create a complete frame set and labeled data [107].

PropBank. Unlike FRAMENET, the primary goal in developing PROPBANK was not the compilation of a lexical resource, but rather the creation of an annotated corpus that could be used to train statistical parsers [108]. In PROPBANK, the difficult annotation of frame evoking elements with abstract global frames is avoided by focusing on a simple scheme of annotating predicate-argument structure with local, verb-specific labels. This means that even highly synonymous verbs ("*seem*" and "*appear*") use different labels. If a verb has more than one broad sense, it may evoke one of multiple verb-specific frames. The PROPBANK project annotated a large portion of the Wall Street Journal with these frame-semantic labels and is considered to be the most comprehensive of all SRL annotation efforts [109]. This data has been used to train statistical SRL systems that predict SRL labels for new sentences [101]. Due to these properties, I focus on PROPBANK-like semantic roles in this thesis.

SRL for other languages. A number of SRL projects exist for only a few languages other than English at varying levels of completeness. PROPBANK-like annotation efforts are under way for Chinese [110], Arabic [32] and Hindi [33]. A FRAMENET-like project exists for German [31], although it is known to be incomplete.

2.4 Summary of Terminology and Notation

I summarize and conclude this chapter by giving an overview of all main terms that were introduced, together with their notation as used in this thesis.

Lexical items All lexical items are given in quotations and italics. If entities are marked in text, they are additionally highlighted bold. An example is "***Albert Einstein** was born in **Ulm***".

Entity mentions and entity types Before Relation Extraction can be executed on a corpus, the first step is to recognize all mentions to entities of interest in text. In this thesis, I regard this step as already completed and use the terms *entity mention* and *entity* interchangeably. Each entity has a *type*, for which I use *smallcaps camelcase* notation. Examples for entity types are thus PERSON, LOCATION, BOOK or CELESTIALOBJECT.

Relation A *relation* is a type of semantic relationship that holds between two entities. Relations are given in *smallcaps camelcase* notation, examples of which are BORNIN and MARRIEDTO. Some relations require additional information to indicate the entity types for which they hold, which are concatenated before and after the relation phrase whenever necessary. For instance, BORNINCOUNTRY is a BORNIN relation that holds specifically for persons born in a country. ENGINEERBORNINCOUNTRY is a BORNIN relation that holds specifically for entities of type ENGINEER and COUNTRY. However, most relations in this thesis are named without entity types.

Relation instances and entity pairs A relation instance is an entity pair that is assigned to a relation. I make no difference in notation, both are written as tuples of italic strings. An example is <*Albert Einstein, Ulm*>.

Patterns Dependency tree patterns are given in quotations and true type. An example is "`[X] and [Y] married`". They contain the placeholders `[X]` and `[Y]` for the subject and object entities in a binary relation. If the entity types of the pattern are restricted, this is included in the placeholders in *smallcaps*. So, "`[X:ACTOR] and [Y:ACTOR] married`" is a dependency pattern that holds only for two entities of type ACTOR.

Semantic Roles Semantic role annotation distinguishes between frame and argument labels. Frame labels indicate which semantic frame is evoked by a verb. They are given in *small-caps*, examples are the frames MARRY.01 or OPEN.01. Argument labels indicate the role of a constituent in a frame. They are given in bold, for example **A1** or **A2**.

For quick reference, there is also a list of abbreviations in the prefix of this thesis. In the next chapter, I examine methods that address the limitation of needing to pre-specify relations for RE and investigate the use of different pattern generation methods for automatic Relation Discovery.

3

Relation Discovery

> *Sheltered by his caste, Sarcellus had not, as the impoverished must, made* fear *the pivot of his passions. As a result he possessed an immovable self-assurance. He felt. He acted. He judged. The fear of being wrong that so characterized Achamian simply did not exist for Cutias Sarcellus. Where Achamian was ignorant of the answers, Sarcellus was ignorant of the* questions. *No certitude, she thought, could be greater.*
>
> – Excerpt from [2]

3.1 Overview

As stated in Chapter 1, the goal is to research methods for Relation Extraction that allow data scientists to work with large corpora of unknown content at little cost. This chapter gives details on the first part of this research, namely the investigation of a fully unsupervised method for *Relation Discovery* (RD). Such methods are able to automatically identify prominent relations in corpora by employing a distributional model of semantics and unsupervised machine learning. I examine the use of an informed linguistic abstraction in this context and create a state-of-the-art Relation Discovery system based on this analysis. Using this system, I evaluate strengths and limitations of fully unsupervised approaches.

This section gives an overview over Relation Discovery and in particular the problems that result from a sparse and ambiguous feature space. In Section 3.1.2, I look at previous work in the field and highlight the shallow pattern generation approaches that had been hitherto used for Relation Discovery. In Section 3.2, I propose a number of pattern generation methods based on typed dependencies and fine-grained type restrictions. Section 3.3 then performs an extensive evaluation in which I compare the proposed pattern generation approaches against earlier baselines on different datasets, and using different clustering setups. The evaluation is conducted both quantitatively against labeled data and qualitatively with the task of Relation Discovery in mind. Section 3.4 concludes this chapter with a discussion on the merits of the proposed approach and the impact on the overall goals of this thesis.

This chapter is based on two previously published full papers, namely [34] and [35], as well as a workshop I organized on this topic [36], but expands on the experimental evaluation and discussion of the proposed approach.

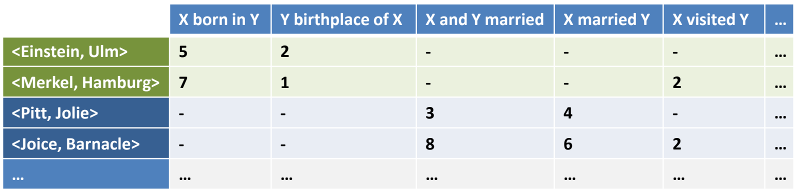
3.1.1 Problem Statement

Recently, there has been great interest in broadening Information Extraction methods to allow for the unsupervised discovery of relational information in large document collections of unknown content [36]. Contrary to classic Information Extraction in which relationship types (such as BORNIN or MARRIEDTO) are specified in advance, such methods automatically identify *a priori* unknown relationship types in a given corpus. For these identified semantic relations, they subsequently or simultaneously perform an Information Extraction step, thereby transforming the corpus into structured, relational data without any supervision or previous knowledge about its content. Given the stated results of this thesis, such approaches may be highly relevant and accordingly are the starting point of my analysis.

Creating a pair-pattern matrix from a corpus. Arguably the most prominent approach in this field, also described as Unsupervised Relation Extraction (URE), addresses this challenge by building on the *latent relation hypothesis* which states that pairs of words that co-occur in similar patterns tend to have similar relations [25, 111]. Current techniques capture this in a vector space model by generating a *pair-pattern matrix* from a given corpus. In this matrix, each row represents an entity pair and each column a distinct pattern. These patterns are extracted from sentences with entity pairs using a supplied *pattern generation* method. The cell values indicate how often each entity pair was observed in the corpus with each pattern. This representation allows us to compute the similarity of two entity pairs by comparing the distribution over observed patterns.

Refer to Figure 3.1 for an example of such a matrix. It shows that <*Einstein, Ulm*> and <*Merkel, Hamburg*> are observed with similar patterns ("`[X] born in [Y]`" and "`[Y] birthplace of [X]`"), indicating that they share the same relation (BORNIN in this case). The same holds for the entity pairs <*Pitt, Jolie*> and <*Joice, Barnacle*> which share the relation MARRIEDTO.

Clustering the pair-pattern matrix. Since we can calculate the distances between entity pairs, clustering methods can be applied to group them into clusters that share similar patterns and can



	X born in Y	Y birthplace of X	X and Y married	X married Y	X visited Y	...
<Einstein, Ulm>	5	2	-	-	-	...
<Merkel, Hamburg>	7	1	-	-	2	...
<Pitt, Jolie>	-	-	3	4	-	...
<Joice, Barnacle>	-	-	8	6	2	...
...	...	...	...	...	...	...

FIGURE 3.1: Example of a pair-pattern matrix. Rows represent entity pairs, columns represent patterns. Cell values indicate how often a pattern was observed for an entity pair in a corpus.

therefore be assumed to represent a relation. This means that ideally, a clustering method over a pair-pattern matrix will return three kinds of structured information, each of which is highly relevant to Relation Discovery and Extraction in corpora of unknown content:

1. **Relations** Each cluster of entity pairs identified using the clustering approach is assumed to represent one discovered relation.
2. **Relation Instances** The entity pairs that make up the clusters are assumed to be the instances of the (binary) relation each cluster represents.
3. **Patterns** For each discovered relation, a set of discriminative *patterns* that extensionally describe the relation may be distilled from the clustering result. Such patterns may be used by a Relation Extraction system to find further *instances* for each discovered relation.

The problem: Ambiguities in a sparse feature space However, in practice, the pair-pattern matrix constructed for a given corpus will be extremely sparse: The space of all possible patterns is typically very large, while there will only be a small handful of observations for most entity pairs. In a scenario of such scant evidence per entity pair, pattern ambiguities may be detrimental.

Ambiguities in patterns may result from the pattern generation approach used for constructing the pair-pattern matrix; Such an approach takes as input a sentence with an entity pair marked in it and outputs one or multiple patterns for the entity pair. Take for instance the five sentences in Table 3.1, each of which has one distinct entity pair (highlighted bold). Entity pairs **(1)** and **(2)** belong to the same relation, namely GRADUATEDWITH (GW), while the other three belong to three distinct relations, namely CONTRACTEDDISEASE (CD), MARRIEDTO (MT) and FRIENDOF (FO). Ideally, the patterns we extract from these sentences would be identical for the first two (thus giving evidence to the clustering method that they belong to the same relation), and distinct for the other three.

However, as can be seen in Table 3.1, depending on the pattern generation method, we get different evidence; A naive approach for example is to simply use the sequence of words between two

Relation	Annotated Sentence	“Between text”	“Between verb”
GW	**(1)** ***Einstein*** *got his* ***PhD*** *in Zurich.*	[X] got his [Y]	[X] got [Y]
	(2) ***Gauss*** *got a* ***PhD*** *in mathematics.*	[X] got a [Y]	
CD	**(3)** ***Dirk*** *got* ***H1N1*** *while on vacation.*	[X] got [Y]	
MT	**(4)*****Joice*** *and* ***Barnacle*** *got married.*	[X] and [Y]	-
FO	**(5)** ***Merkel*** *and* ***Hollande*** *became friends.*		-

Table 3.1: Five sentences with one distinct entity pair each (highlighted bold). The entity pairs belong to four different relations (the first two sentences belong to the same relation). Depending on how we observe patterns, evidence points to different clusters.

entities in a sentence as pattern (the “between text” column in Table 3.1). However, as we can see this results in two *distinct* patterns for entity pairs **(1)** and **(2)**. Worse, this method finds the *same* pattern for entity pairs **(4)** and **(5)**, giving false evidence that they share the same relation.

A second naive way of extracting patterns would be to use only a verb if it occurs between to entities as pattern (the “between verb” column in Table 3.1). However, as can be seen in Table 3.1, while this correctly finds the same pattern for entity pairs **(1)** and **(2)**, it now incorrectly includes entity pair **(3)**, which has a different relation. Also, this way no evidence at all is found for entity pairs **(4)** and **(5)**. This example illustrates that in order for Relation Discovery to work, the pattern generation method must be constructed in such a way that ambiguities are minimized, while at the same time not overspecifying patterns.

For the purpose of pattern generation, I make a distinction between two components of patterns:

1. **A lexico-syntactic component** The first component is the *lexico-syntactic pattern* that holds between two entities in a text.

2. **A type restrictions component** The second (optional) component are *type restrictions* on the entities in a pattern. Such restrictions are necessary since lexico-syntactic patterns alone may not be discriminative enough in many cases. Consider entity pairs **(3)** and **(1)** in Table 3.1: In both cases, the relation is mediated by the verb “*get*”. The only way to distinguish between the two is by knowing that the entities “PhD” and “H1N1” are of different types.

Entity types in the open domain. Especially the second component of pattern generation presents a problem considering our goal of Relation Discovery in the open domain. Here, one may encounter a potentially unrestricted set of entities of arbitrary types and granularity that varies from corpus to corpus. For example, the types of a standard NER tagger (Person, Location, Organization etc.) may be too coarse-grained for the above example, not being able to distinguish between Disease and Degree. While more fine-grained NER taggers have recently been researched [112], it is unclear whether they can be applied to the open domain.

In this chapter, I address the issue of ambiguities in a sparse feature space by investigating the impact of pattern generation on Relation Discovery. I examine both the lexico-syntactic as well as the type restriction aspects of patterns and propose solutions that reduce ambiguities while not overspecifying patterns.

3.1.2 Related Work

I review previous work in Relation Discovery with regards to pattern generation and and identify evaluation baselines (Section 3.1.2.1). I then give an overview of other NLP tasks commonly addressed with clustering and distributional semantics (Section 3.1.2.2).

3.1.2.1 Pattern Generation for Relation Discovery

Lexical patterns with skips (BASELINE-TUR). [25] cluster entity pairs in the pair-pattern matrix to identify semantic relations. The resulting clusters are interpreted as each representing one relation that holds between all entity pairs in the cluster. They use the text between two entities in a sentence as patterns, but also allow arbitrary word skips, meaning that for each sentence containing an entity pair a large number of features are generated. The same pattern generation method is also used by [90] albeit for a slightly different task, namely to solve the problem of finding analogies between word pairs.

I use this method as a baseline; for each entity pair in a sentence, I first determine the words that are between them and then build the power set (the set of all possible subsets) of all "between words". For each subset, I then generate a pattern in which the words in the subset are skipped. This means that there is a theoretical maximum of 2^n patterns, where n is the number of between words. In order to reduce the size of the feature space, I only consider entities with at most 8 "between words". For the remainder of this chapter, this baseline is referred to as BASELINE-TUR.

Subsequences including pre- and postfix spans (BASELINE-BOL). [26] propose a co-clustering approach that simultaneously clusters both entity pairs and patterns for identifying relations, using not only lexical, but also shallow syntactic patterns. They expand the previously discussed pattern generation approach to also include prefix and postfix spans, i.e. the words that come before and after the entity pair in a sentence. In addition, they place limitations on word skips and the total length of the pattern. They use three variables: L is the maximum number of words that may occur in a subsequence, g is the maximum number of consecutive words that may be skipped, while G is the maximum number of words that may be skipped in total. They determine these values experimentally and set them to $L = 5$, $g = 2$ and $G = 5$ for their evaluation.

I use this approach as a second baseline, and use the same parameterization [26] used in their experiments. For the remainder of this chapter, this baseline is referred to as BASELINE-BOL.

Lexical patterns with entity type restrictions (BASELINE-WAN). More recently, [91] analyzed the impact of filtering techniques and found that overall clustering quality f-measure significantly increases by using a set of filters to eliminate patterns that are unlikely to represent a relation. They filter out a total of 80% of all observed patterns. They use the text between entities as patterns, without word skips, and include information from NER taggers into the feature set. They filter patterns with more than ten between words and more than 1 distinct verb. I use a reimplementation of this approach as the third baseline, hereafter referred to as BASELINE-WAN.

Topics and fine-grained Named Entity types. [113] use a very rich feature set, including fine-grained Named Entity types and document topics, to first disambiguate each pattern individually and in a second step perform RD using disambiguated patterns. However, this approach requires a massive redundancy of pattern observations for disambiguation; in their experiments, they handled only patterns that are seen more than 200 times in their corpus. As such redundancy is unlikely to occur, the approach is impractical for most corpora and therefore does not serve as a baseline. However, I use the idea of fine-grained Named Entity types as a baseline for entity type restrictions in patterns. These baselines are introduced in detail in Section 3.2.2.

Contrary to previous approaches in Relation Discovery at the time of investigation, I employ a pattern generation technique that utilizes information from a dependency parser and propose to use either word clusters or fine-grained NE types as entity type restrictions. The observation is that current dependency parsers are becoming orders of magnitudes faster while retaining a sufficiently high precision and recall (see [114] and [115]), and that both supervised as well as unsupervised methods for modeling fine-grained entity types in the open domain are becoming possible. I comparatively evaluate the proposed pattern generation techniques against baselines modeled after the approaches mentioned above.

3.1.2.2 Distributional Semantics for Other Tasks

The latent relation hypothesis is an application of *distributional semantics*, which characterizes the semantics of a linguistic item (such as an entity pair, a word or a pattern) by co-occurring linguistic items. This observation has roots in early work by the American linguist Zellig Sabbettai Harris' work on distributional structure [116], as well as Austrian philosopher Ludwig Wittgenstein's observation that meaning is associated with use ("*die Gebrauchstheorie der Bedeutung*") and therefore observable [117]. With increasing capability for NLP researchers to process ever larger amounts of data, distributional semantics is now being used to address a variety of tasks next to Relation Discovery, some of which are relevant for the research presented in this thesis.

Similarity of patterns. Instead of using clustering to identify relations, some previous work has investigated measuring the pairwise similarity of patterns. [90] compute the pairwise similarity of lexical patterns to solve the problem of finding analogies between word pairs. [118] compare pairs of words using the distribution over patterns to find proportional analogies and evaluate this on corpora of word comprehension tests, such as analogy questions in SAT or TOEFL tests. By contrast, [119] directly measure the pairwise similarity between patterns in dependency trees using the distribution over word pairs to find inference rules from text. [78] extend this with a clustering approach to group patterns into clusters, which they use to guide semi-supervised Relation Extraction methods. Patterns in each cluster may then be interpreted as paraphrases, although the clustering they use is "hard", meaning that each pattern is assigned to exactly one cluster. This runs contrary to the intuition that each pattern may give different amounts of evidence to different semantic relations. Nevertheless, this shows that simply by *transposing* the pair-pattern matrix, we can measure the similarity of patterns and group them into paraphrase clusters. The research presented in Chapter 4, which builds on the results of this chapter, makes use of this idea.

Similarity of words. One of the first applications of distributional semantics was to measure the similarity of words using co-occurring words in a corpus [120]. This idea was epitomized by British linguist John Rupert Firth who famously said "*You shall know a word by the company it keeps*" [121]. Previous work has used distributional semantics to produce clusters of words assumed to share semantic properties. Since words are often ambiguous and may have different semantic properties, multiple or overlapping clusterings are often produced, grouping words into more than one cluster. A large-scale example of this is a clustering of more than 10 million distinct one-to-five-word-grams from the Google n-gram data set [122] computed by [123]. In this *phrasal clustering*, each phrase is clustered into up to 10 distinct clusters with different confidence values. Previous work has leveraged the latent semantic information given by phrasal cluster memberships of n-grams to solve tasks other than URE. For example, [124] increase the performance of deep syntactic parsers with regard to long-range dependencies, and [125] transfer linguistic structure using cross-lingual word clusters.

One of the ideas I investigate in this chapter is to avoid using a manually constructed type system for entity restrictions, arguing that in the open domain such pre-construction of a type system might be too limiting. I instead investigate the use of a phrasal clustering for determining entity type restrictions, by interpreting each cluster as an entity type and all n-grams assigned to a cluster as belonging to this type. I incorporate this into the pattern generation step of the RD method and use this information to model type restrictions. Thus, the type system is not manually specified, but rather induced without supervision from a large Web corpus, making it a natural fit for the open domain and RD.

3.1.3 Contributions

I propose several informed pattern generation approaches for use in Relation Discovery and comparatively evaluate them against a set of baseline approaches. Based on this, I create a state-of-the-art Relation Discovery system which I use to evaluate strengths and limitations of fully unsupervised approaches for handling unknown corpora. In more detail, the contributions are:

Algorithm for pattern generation in a dependency tree. I propose to address the problems of pattern ambiguities and overspecification with a pattern generation approach that utilizes dependency trees. I present two variants of the approach: First, an algorithm that selects possible patterns for a given entity pair in a dependency path, as an extension of the *shortest path* method. The approach is capable of capturing a wider range of phenomena than previous part-of-speech based pattern generation and filtering approaches by incorporating syntactic elements for long range dependencies, complements for light or support verbs, appositions and context for arguments in direct conjunction. Second, I present a variant in which all possible subtrees are generated from dependency trees and used as patterns. I show that the best proposed methods increase the clustering quality f-measure by up to 28 percentage points over the best baseline approaches.

Method for modeling type restrictions for Relation Discovery in the open domain. I propose a novel method that leverages a Web-derived clustering of n-grams to model restrictions in the open domain. Contrary to previous approaches, it is unsupervised and thus does not require a pre-specification of entity types. I compare the proposed approach against baselines in which I model type restrictions using the Stanford NER tagger [126] as well as fine-grained Named Entity classes derived from the YAGO knowledge base [83]. The comparative evaluation shows that more informed entity type restrictions using YAGO and the proposed method significantly improve Relation Discovery.

Discussion of unsupervised Relation Discovery. I discuss limitations of fully unsupervised approaches for Relation Discovery and argue, based on the results of qualitative evaluations, that some supervision is necessary *1)* to correct errors from misleading distributional evidence and *2)* to direct discovery towards a user-defined information need. This is investigated more closely in Chapter 4.

3.2 Pattern Generation for Relation Discovery

Pattern generation takes as input a set of sentences in which each sentence contains at least two entities of interest. For each entity pair, one or more patterns are generated from each sentence. As discussed in Section 3.1.1, the main challenge is to generate patterns in such a way as to avoid both over- and underspecification. In this section, I investigate the two aspects of patterns in this regard: Their lexico-syntactic aspect (Section 3.2.1) and their entity type restriction aspect (Section 3.2.2).

3.2.1 Lexico-Syntactic Patterns

I propose a pattern generation approach that utilizes dependency trees[1] to generate a list of patterns for each sentence and entity pair. I propose two variants:

3.2.1.1 Extended Shortest Path (PROPOSED-ESP)

The first proposed pattern generation approach is the *extended shortest path* (hereafter referred to as PROPOSED-ESP). Here, first a set of *core tokens* is determined by collecting all tokens on the shortest path that connect the two entities in a dependency tree. The shortest path is then extended by finding a set of *optional tokens* linked to a core token with certain typed dependencies. The approach then generates one pattern for each combination of the core tokens and the power set (the set of all possible subsets) of the optional tokens.

Typed dependencies that indicate possibly important information even if not on the shortest path were determined through experimentation. Simple examples of cases in which important information is not on the shortest path are negations and particles, which are directly connected to a verb (with the dependencies "*neg*" and "*prt*" respectively) but never function as a link on the path between two arguments bound by this verb. Other examples are appositions, which may be connected to an entity but are not themselves part of the shortest path (indicated by "*nn*" or "*appos*"), and light verb constructions in which only the verb, but not the typically more important noun is part of the shortest path. Another example - discussed in detail below - are two entities in conjunction that function as an argument for a verb.

The method consists of four steps:

Step 1: Compute the shortest path between subject and object. The shortest path between two entities in a dependency path serves as basis for our extraction method. By focusing on the tokens that syntactically link both entities, we can skip over tokens that are less likely to be relevant to the relationship [92, 128]. This step yields a list of core tokens likely to be relevant to the relation expressed between the two entities.

Step 2: Collect a set optional tokens on the path. Collect all tokens that *may* be relevant to identifying a relation by iterating over each token on the shortest path and examining all typed dependencies of each token to non-path tokens. If the dependency is one of {*nn*, *neg*, *prt*, *poss*,

[1] In this section, I use Stanford typed dependencies [127] but other sets of typed dependencies could also be used.

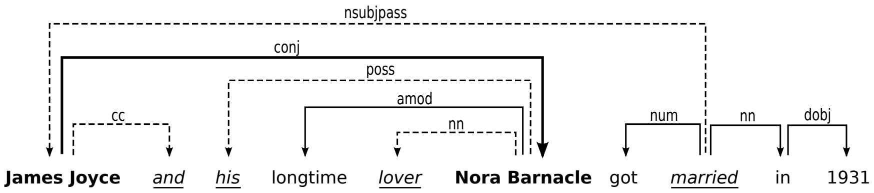


FIGURE 3.2: Dependency parse of the example sentence. The entity pair and shortest path are marked in bold. ”*James Joyce*“ and ”*Nora Barnacle*“ are directly connected with a ”*conj*“ link. Links to optional tokens are illustrated as dotted lines; optional tokens are underlined.

possessive, *nsubj*, *nsubjpass*} we collect the target token into a list of optional tokens. This step yields a list of tokens to be added to the core list to produce a good extraction pattern.

Step 3: Generate patterns. Build the power set over all optional tokens and generate one pattern for each combination of the shortest path and optional set. This power set includes the empty set as well, so the shortest path without any optional tokens is included in the patterns.

Step 4: Remove uninformative patterns. Filter out all patterns that consist only of closed-world word classes. Examples are patterns like “`[X] and [Y]`”. The intuition for this step is that such patterns are semantically too weak to be used as patterns and not suitable for clustering approaches.

The following example sentence illustrates the pattern generation process: *“**James Joyce** and his longtime lover **Nora Barnacle** got married in 1931”*. Figure 3.2 depicts the sentence’s dependency parse. Here, the shortest path is a “*conj*”-link, directly connecting the two entities ”*James Joyce*“ and ”*Nora Barnacle*“. The resulting pattern ”`[X] and [Y]`“[2] is highly ambiguous and therefore of limited use. We collect the tokens ”*and*“, ”*his*“, ”*lover*“ and ”*married*“ into a set of optional tokens and build its power set. By taking each combination of the power set and the shortest path (and after filtering non-informative features) we arrive at a total of five features, as listed in Table 3.2.

3.2.1.2 All Subtrees (PROPOSED-SUB)

The second proposed method (hereafter referred to as PROPOSED-SUB) is to generate a pattern from *any* subtree in a dependency tree, as long as it is connected and spans the two entities of interest. Since all patterns generated by the extended shortest path approach are also subtrees, this variant produces a superset of the patterns generated by PROPOSED-ESP. A possible advantage of this approach is that it does not require any manual selection of interesting subtrees, unlike

[2] In this case, the pattern ”`[X] and [Y]`“ is a verbalization of the entities [X] and [Y] being linked by the typed dependency ”*conj*“ for readability reasons.

BASELINE-TUR	[X] and his longtime lover [Y], [X] and his longtime * [Y], [X] and his * lover [Y], [X] and * longtime lover [Y], [X] * his longtime lover [Y], [..]
BASELINE-WAN	[X:PERSON] and his longtime lover [Y:PERSON]
PROPOSED-ESP	[X] and lover [Y], [X] and [Y] married, [X] and lover [Y] married, [X] and his lover [Y], [X] and his lover [Y] married
PROPOSED-SUB	[X] and lover [Y], [X] and [Y] got married, [X] and his longtime [Y], [X] and [Y] got married in [..]

TABLE 3.2: Patterns from different generation methods for the sentence in Figure 3.2.

PROPOSED-ESP which requires a specification of links that indicate optional tokens. However, a possible disadvantage is that this approach generates a much larger number of patterns per entity pair, some of which may be highly ambiguous.

Table 3.2 shows patterns generated for the sentence presented in Figure 3.2 with the two proposed approaches and two baselines. As can be seen, the baselines are not effective in generating pattern; With BASELINE-TUR, which generates purely lexical patterns and allows arbitrary word skips as indicated by the asterisk, a large number of patterns are generated, many of which are underspecified. On the other hand, with BASELINE-WAN only one pattern is generated using the entire string between the two entities and entity type restrictions. This pattern is overspecified with regards to the relation; it is unlikely to occur often in a given corpus and may thus be of little use as evidence for a clustering approach. PROPOSED-ESP strikes a balance between the two by identifying important words based on deep syntactic information and generating patterns from their permutations. PROPOSED-SUB also generates these patterns, but additionally finds other subtrees, some of which are more relevant while others are redundant.

3.2.2 Entity Type Restrictions

As mentioned in the introduction to this chapter, even with the pattern generation method proposed in the previous section, ambiguities remain that can only be resolved at the entity level. Recall the opening example from Section 3.1.1 in which the same lexico-syntactic pattern, namely ”[X] get [Y]“, is observed in different contexts; once between a person and

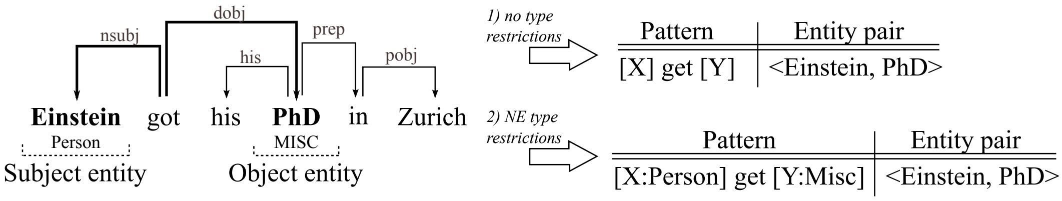


Figure 3.3: Illustration of the pattern generation process for one example sentence with the entities "Einstein" and "PhD". Named Entity class tags are below the tokens in the sentence, the dependency tree above with the shortest path highlighted bold. The pattern is generated once with and without NE classes as restrictions.

a degree, indicating the GraduatedWith relation, and once between a person and a disease, indicating the ContractedDisease relation. In this section, I propose several methods for including entity type restrictions into patterns in order to make them distinctive in such cases.

3.2.2.1 Named Entity Type Restrictions

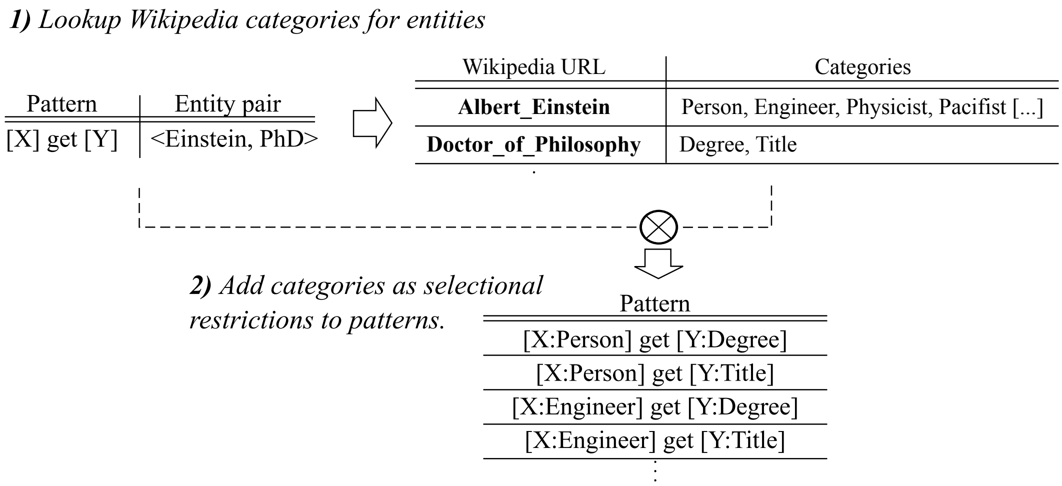
The first extension of the proposed method is straightforward: Simply include standard Named Entity types as restrictions, in a similar fashion as in previous approaches [129]. I incorporate the Stanford NER 7-class tagger [126] into the sentence parsing pipeline and determine the type of each entity. These types are used to restrict the generic placeholders `[X]` and `[Y]` in generated patterns with the types of the subject and object entities.

For the example sentence illustrated in Figure 3.3, the tagger determines the class Person for "Einstein", and Misc for "PhD". The latter class is used for all entities that cannot be assigned any of the named classes. We therefore generate the pattern "`[X:Person] get [Y:Misc]`" in this example. Because we model entity type restrictions directly into the patterns, we increase the space of all possible patterns and make individual patterns more discriminative.

Limitations. However, as shown in the example in Section 3.1.1, the Named Entity classes given by a 7-class tagger are coarse grained and may not include the types necessary to disambiguate all patterns. Also, there is a risk that Named Entity taggers may determine the wrong type for an entity. For instance, [126] report an overall f-measure of 87% on the CoNLL 2003 Named Entity Recognition dataset. This could lead to false evidence that negatively impacts RD.

3.2.2.2 Fine-grained Entity Type Restrictions

Because classes from a standard 7-class NER tagger may be too coarse grained for RD, I next extend the system with the option of modeling fine-grained Named Entity classes. I choose an


1) Lookup Wikipedia categories for entities
2) Add categories as selectional restrictions to patterns.

Pattern	Entity pair
[X] get [Y]	<Einstein, PhD>

Wikipedia URL	Categories
Albert_Einstein	Person, Engineer, Physicist, Pacifist [...]
Doctor_of_Philosophy	Degree, Title

Pattern
[X:Person] get [Y:Degree]
[X:Person] get [Y:Title]
[X:Engineer] get [Y:Degree]
[X:Engineer] get [Y:Title]

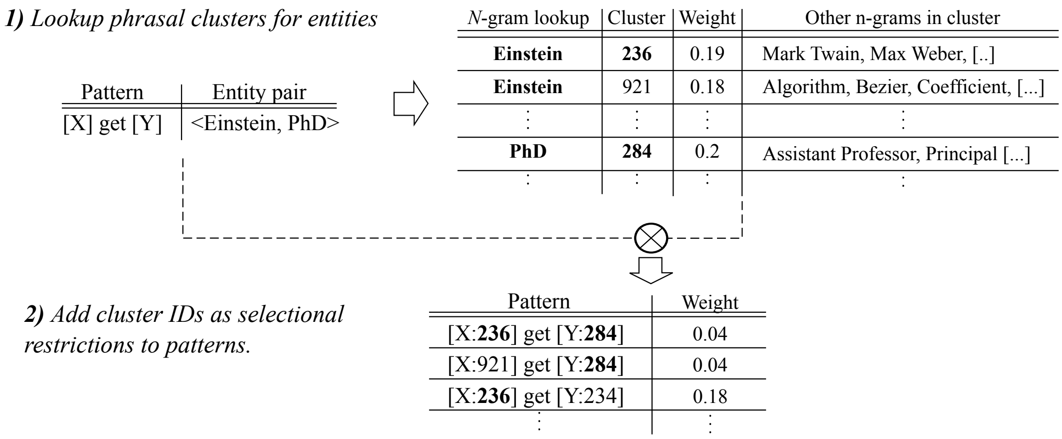
FIGURE 3.4: Illustration of the using Wikipedia categories as type restrictions. In 1), entities are linked to Wikipedia and their categories retrieved. In 2) the Cartesian product over the categories for subject and object is built and used as type restrictions for the pattern. This yields a set of patterns with different type restrictions.

approach that requires entities to be disambiguated and linked to Wikipedia URIs. The YAGO knowledge base then enables the system to retrieve fine-grained entity classes for disambiguated entities, such as their Wikipedia categories (of which I use only the head nouns as restrictions). Because many YAGO entities belong to more than one class, this method returns a set of classes for each entity. For example, the two Wikipedia categories for "PhD" are "DOCTORAL DEGREES" and "TITLES", while the Wikipedia page for "Albert Einstein" is in over 50 categories[3] I use only the head noun of each category, so "DOCTORAL DEGREES" is shortened to the type DEGREE.

For each entity pair, I retrieve two sets of entity classes (one for the subject and one for the object). I determine the Cartesian product over these two sets and create one distinct pattern with type restrictions for each combination. For the example sentence, this means that I generate a list of patterns, including "`[X:Person] get [Y:Degree]`", "`[X:Person] get [Y:Title]`" and "`[X:Engineer] get [Y:Degree]`", each of which is used as a pattern. Refer to Figure 3.4 for an illustration of this. While this method increases the overall number of observed patterns by about one order of magnitude, individual patterns are much more discriminative than without type restrictions.

Limitations. Two things must be noted regarding this method of determining fine-grained Named Entity classes. Firstly, it does not necessarily produce patterns at the desired granularity. For instance, the distinction of types PHYSICIST and ENGINEER for the entity "Einstein" may be too fine grained for most cases where it is sufficient to know that the entity is of type PERSON. More importantly though, the method is limited to entities that can be disambiguated

[3] http://en.wikipedia.org/wiki/Albert_Einstein, accessed 03/21/2015.


1) *Lookup phrasal clusters for entities*
2) *Add cluster IDs as selectional restrictions to patterns.*

Pattern	Entity pair
[X] get [Y]	<Einstein, PhD>

N-gram lookup	Cluster	Weight	Other n-grams in cluster
Einstein	**236**	0.19	Mark Twain, Max Weber, [..]
Einstein	921	0.18	Algorithm, Bezier, Coefficient, [...]
⋮	⋮	⋮	⋮
PhD	**284**	0.2	Assistant Professor, Principal [...]
⋮	⋮	⋮	⋮

Pattern	Weight
[X:**236**] get [Y:**284**]	0.04
[X:921] get [Y:**284**]	0.04
[X:**236**] get [Y:234]	0.18
⋮	⋮

FIGURE 3.5: Illustration of the proposed pattern generation process that uses phrasal cluster memberships as type restrictions.

to an appropriate Wikipedia page. While this is possible on the dataset I use for the evaluation, it is much more difficult to determine fine-grained Named Entity classes in the open domain with this method. This potentially limits the usefulness of this method for Relation Discovery.

3.2.2.3 Phrasal Clusters as Restrictions

To address these limitations, I propose a method for modeling type restrictions that does not require an existing type system or the disambiguation of entities.

I extend the system with the option of using type restrictions derived from a phrasal clustering computed by [122] over a dataset of more than 10 million distinct one-to-five-word-grams from the Google n-gram data set [122]. In this dataset, each n-gram is assigned to ten different phrasal clusters with different association values, also referred to as *weights*. Weights are values between 0 and 1, with a higher value indicating a stronger assignment confidence. Because the clustering is based on lexical context, n-grams in a cluster often share semantic properties. For example, the dataset contains clusters of entities like cities, cars, movies, etc [122].

During pattern generation, the method looks up the phrasal cluster IDs for the lexical representation of an entity and uses this ID as type restriction. For example, the string “*Einstein*” belongs to phrasal cluster number 236 with weight 0.18. Semantically similar strings, denoting person names such as “*Max Weber*” and “*Marc Twain*” are also part of this cluster. I can use this information to restrict the subject of the pattern only to strings that belong to cluster 236. Another phrasal cluster for “*Einstein*” is cluster 921 (with a lower weight of 0.17), which contains more general terms from mathematics such as “*Algorithm*”, “*Bezier*” and “*Coefficient*”. “*PhD*” is found in cluster 825, which contains academic titles and positions such as “*Assistant Professor*” and “*Principal*”. See Figure 3.5 for an illustration of this example.

I build the Cartesian product over the two sets of phrasal clusters retrieved for the subject and object of an entity pair. Because each entity (e.g. its lexical representation) has 10 soft cluster memberships, the Cartesian product of phrasal clusters for both entities of an entity pair yields a total of 100 distinct weighted phrasal cluster ID combinations, hereafter referred to as *restriction pairs*. The weight of each restriction pair is computed by building the product of the confidence weights of the respective entity-phrasal cluster assignments. Each restriction pair is encoded into its pattern by adding to the entity placeholders "`[X]`" and "`[Y]`" a qualifier indicating the phrasal cluster ID. For each observation and restriction pair, a distinct pattern is generated.

This option increases the overall number of distinct patterns by two orders of magnitude. Patterns are also less humanly readable than their counterparts that use coarse- or fine-grained Named Entity types. I use this feature space to evaluate the assumption that one can leverage distributional evidence over a large Web corpus to model type restrictions in RD without an existing type system.

3.3 Evaluation

I evaluate the proposed pattern generation approaches in a series of experiments with different datasets and clustering setups. Each set of experiments is followed by a qualitative discussion that motivate the next set. I begin this section with a detailed discussion of the evaluation setup and the many challenges of evaluating discovery approaches and clustering in general.

3.3.1 Experimental Setup

The principal challenge in evaluating clustering approaches for Relation Discovery are the number of components that each potentially heavily impact evaluation results, and the prohibitively large number of possible permutations of evaluation setups that are theoretically possible. I identify the following five components in the pattern extraction and clustering pipeline: *1)* The size and composition of the evaluation dataset. *2)* The pattern extraction and filtering method. *3)* An optional step of assigning weights to patterns based on certain criteria. *4)* The clustering approach and its parameterization. *5)* The evaluation measures. In the following, I discuss each of these components, the difficulties they pose for evaluation and describe the approach that I took.

3.3.1.1 Dataset

Difficulties One of the principal questions is the issue of how to find or generate large amounts of gold standard data for the task of Relation Discovery. Ideally, such a dataset would be large

sentence retrieved	**relation expressed**
Mystery Men *(1999) stars* ***Ben Stiller*** *as Mr. Furious.*	explicit
Mystery Men *brought on board a talented cast from William H. Macy to* ***Ben Stiller****.*	explicit
What was ***Ben Stiller****'s character's super quality in* ***Mystery Men****?*	implicit
Ben Stiller *does not think* ***Mystery Men*** *should be remade.*	false

TABLE 3.3: Sentences retrieved for the entity pair <*Ben Stiller*,*Mystery Men*>, labeled as ACTEDIN in YAGO, and the degree of how explicitly the relation is expressed: explicit, implicit or not at all.

and have gold standard relation annotations for each contained entity pair. Previous projects have constructed such a ground truth manually [25], which has a number of drawbacks: Firstly, there is a high cost involved in manually annotating sentences with relations, limiting the size of the ground truth as well as the ability to quickly generate new evaluation sets. Secondly, much care must be taken to ensure that no approach-specific assumptions are modeled into the ground truth, i.e. "overfitting" the ground truth to the capabilities of the algorithm that is to be evaluated. The inherent risk in manual annotation is the creation of a ground truth that does not realistically reflect the application scenario the Relation Discovery approach is intended for.

Chosen Approach Because of these difficulties, I chose an approach that allows me to automatically generate *silver standard* evaluation datasets of different sizes and compositions. A silver standard is a dataset that is automatically labeled using data from an existing knowledge base of facts (triples consisting of two entities and a relation that holds between the entities) using *distant supervision* [12] (see Section 2.2.3.2). I used YAGO, a semantic knowledge base derived from Wikipedia, WordNet and GeoNames with knowledge of more than 10 million entities and around 447 million facts [83] to provide supervision. Using a set of parameters, we sampled entity pairs from YAGO and for each entity pair retrieved a set of sentences from the Web that make mention to them[4]. The distant supervision assumption is that a sentence that contains an entity pair for which the KB specifies a relation is likely to express it, either explicitly or implicitly. Accordingly, this allows the method to automatically label all retrieved sentences with relations, enabling the generation of a ground truth of arbitrary size.

However, there is no guarantee that a sentence containing a specific entity pair in fact expresses the relation as specified by the knowledge base. To gain insight into the strength of the assumption, in [34] we manually examined 200 sentences retrieved and labeled with this approach[5].

[4] My student Do Tuan Anh used the BING API (`http://www.bing.com/developers/`) to create this dataset as part of his Bachelor thesis [130].

[5] This investigation was mostly conducted by my colleague Holmer Hemsen.

Dtaset	Sampling	# Relations	# Sentences
R10-ZIPFIAN	Uniform	10	1,000,000
R10-ZIPFIAN	Zipfian	10	1,000,000
R20-UNIFORM	Uniform	20	1,000,000
R20-ZIPFIAN	Zipfian	20	1,000,000

TABLE 3.4: The four evaluation datasets created using YAGO and distant supervision. The datasets have either 10 or 20 distinct relations. Entity pairs were sampled either randomly, resulting in a zipfian distribution over relations, or uniformly.

They contain a total of 209 relations and 29 distinct relations[6]. In 159 cases the relation is either explicitly or implicitly represented in the sentence, whereas in 50 cases the entity pair is present in the sentence but the YAGO relation between them could not be inferred from the text.

Examples for explicit, implicit and false sentences are given in Table 3.3. While imperfect, the assumption therefore holds for approximately 76% of the generated ground truth. For the purpose of evaluation I find this satisfactory, as this realistically simulates noise while reliably indicating the relational content of generated evaluation sets. Also, since each entity pair typically appears in several sentences in the ground truth, the chance is reasonably high that at least one of the sentences will make mention of the annotated relation.

Generated Silver Standard Datasets I use this approach to generate 4 different silver standard datasets for the experiments. Each of these contains approximately 100.000 sentences. In order to determine how the approach handles differently composed datasets, I vary the amount of distinct relations contained in the datasets: Two contain 20 distinct relations, while two contain 10 distinct relations. For additional experiments, I also vary the distribution of how entity pairs are sampled: In the datasets marked "uniform", each relation has approximately the same number of instances. In the datasets marked "zipfian", the entity pairs are randomly sampled, leading in effect to a zipfian distribution dominated by a few prominent relations and a long tail of more rare relations. Refer to Table 3.4 for a list of all datasets.

3.3.1.2 Pattern Generation

This is the crucial step in the clustering pipeline which I evaluate. Patterns consist of lexico-syntactic patterns plus optional entity type restrictions.

Lexico-syntactic pattern. I use three baseline approaches for patterns: The first, BASELINE-TUR, is based on [118] and [25] and uses a lexical pattern generation technique with arbitrary word

[6] In 9 cases, one entity pair has more than one relation in YAGO. Some persons, for example, both ACTEDIN and PRODUCED a movie.

skips. The second, BASELINE-BOL, is modeled after [26], uses shallow lexico-syntactic patterns including pre- and postfix spans and allows limited word skips. The third, BASELINE-WAN, uses lexical patterns without word skips, applies filtering rules and incorporates Named Entity class information as type restrictions. All three baselines were introduced in detail in Section 3.1.2.1.

I compare the two proposed approaches for generating lexico-syntactic patterns using dependency trees against these baselines. As illustrated in Section 3.2, there are two flavors of the proposed approach: The first is the extended shortest path which focuses on tokens along the shortest path in a dependency tree plus a manually determined set of optional tokens (PROPOSED-ESP). The second is to use arbitrary subtrees in dependency trees that span the two entities of interest (PROPOSED-SUB). The main difference between the two approaches is that manual work is invested in the former to identify good subtrees, while the latter simply takes all possible subtrees. The evaluation will therefore determine *1)* whether deep syntactic information outperforms the shallow baselines, and *2)* whether investing manual work to identify good subtrees pays off in terms of Relation Discovery quality.

Entity type restrictions. I evaluate three baseline approaches for modeling type restrictions: The first, **NONE**, is a setup in which there are no type restrictions at all. The second, **NER** is a setup which, like previous work by [129], uses a standard NER tagger to model type restrictions (see Section 3.2.2.1). In addition to these basic setups, I also evaluate a highly informed baseline in which a high quality, fine-grained type system is used to model type restrictions. In this setup, denoted as **YAGO**, I use fine-grained entity classes derived from Wikipedia categories as described in Section 3.2.2.2.

I compare these approaches against the proposed method of using phrasal clusters as type restrictions, in three parameterizations: The first, **PHRASAL-1**, is a modification of the proposed method in which I only use the cluster with the top weight (instead of all 10) as restriction for an entity. Similarly, the second, **PHRASAL-5**, uses only the top 5 clusters for each string as restrictions. I use this setup to assess the impact of using only the most likely portion of the full phrasal clusters data set. Finally, **PHRASAL-FULL** is the setup in which I use full phrasal clusters data set. The evaluation will therefore determine *1)* whether phrasal clusters or fine-grained NE types improve Relation Discovery, and *2)* whether the full phrasal clustering dataset is necessary for optimal performance.

Since the main purpose of the evaluation is to determine the impact of the pattern generation method on RD, the above options are a varied across all experiments.

3.3.1.3 Feature Weighting

Difficulties Another step in the pipeline with a high potential impact on overall results is the question on how to assign weights to extracted features. While the baseline approach is to simply use occurrence counts as feature weights, hypothetical options include manually assigning different weights to different classes of features. For instance, certain types of deep syntactic patterns between two entities might be judged to be more indicative of relations, while others might generally be more ambiguous. In addition, there is a family of tf-idf [131, 132] schemes that automatically assign weights to features based on the significance of their correlation to entity pairs.

Approach Chosen In order to assess the impact of feature weighting, I compare two baseline methods in the experiments: The first is the above mentioned baseline of simply using co-occurrence counts as weights. The second is to use the positive Pointwise Information Measure (pPMI, explained below) [133], which has been shown to outperform a wide range of other reweighting methods for the purpose of measuring semantic similarity [134]. This approach was first used for the pair-pattern matrix by [111]. I evaluate each pattern generation method with both feature weighting schemes in order to measure their impact.

Positive Pointwise Mutual Information The reweighting method used is a variant of Pointwise Mutual Information [135] (PMI) which is used to determine the statistical significance of a co-occurrence by comparing actual co-occurrence against an expected value estimated from occurrence statistics. Let $p(f_i)$ be the probability of a feature f_i appearing in the corpus, calculated as the number of observations of f_i divided by the total number of observed features. Similarly, let $p(ep_j)$ be the probability of an entity pair ep_j appearing in the corpus, calculated as the number of observations of ep_j divided by the total number of observed entity pairs. If both are independent, then the probability of both occurring together is $p(f_i)p(ep_j)$. This is compared against the observed probability of co-occurrence $p(f_i, ep_j)$. The PMI is thus calculated as:

$$PMI(f_i, ep_j) = \log\left[\frac{p(f_i, ep_j)}{p(f_i)p(ep_j)}\right] \tag{3.1}$$

A PMI above 0 indicates a significant correlation, while a value below 0 indicates a spurious correlation. Since the negative values are not of interest, the positive PMI (pPMI) sets all negative values to 0.

3.3.1.4 Clustering Approach and Parameterization

Difficulties Perhaps one of the most difficult problems for the evaluation is presented by the plethora of clustering approaches that exist, each of which has one or multiple parameters that must be estimated or manually set. Since clustering algorithms have differing underlying philosophies, the choice of clustering algorithm and its parameters may greatly influence evaluation results. However, an exhaustive evaluation of all clustering approaches and parameters is not possible within this chapter.

Approach Chosen In the evaluation, I use k-means, the arguably most commonly known clustering approach which is used to partition a set of points (entity pairs in this case) into k clusters [136]. Each cluster is represented by a *centroid* that is calculated as the *mean* of all points in the cluster. The clustering approach works iteratively, in a "hard" expectation-maximization fashion: In the beginning, k centroids are provided or randomly seeded. At each iteration all points are assigned to their nearest centroid (e.g. the "expectation" step). The centroids are then recomputed based on the points that are assigned to them (the "maximization" step). When no more points change cluster affiliation between two iterations, the approach converges and returns the clustering result. Since all points are assigned to their nearest cluster, the approach effectively splits the dataset into Voronoi cells. Following [134], I use the Cosine similarity to measure distances between feature vectors – a measure useful for highly sparse vectors like the ones at hand.

The k-means algorithm requires us to manually specify the k, i.e. the number of clusters we want to partition the dataset into. For unknown datasets, this means that we need to estimate the k which may be considered suboptimal. However, it may be argued that through this parameter, we can also influence the *granularity* of discovered relations: If we are only interested in broad partitions, we run k-means with a low k, while if we are interested in obtaining a large number of fine-granular relations, we run it with a high k. By experimenting with *parameter sweeps* over a large range of k, I seek to determine whether this intuition holds true.

3.3.1.5 Evaluation Measures

Difficulties Given gold standard labels, the evaluation of clustering has a set of standard measurements, as explained below. However, some difficulties remain: The first is how to treat *unclustered* points, i.e. points that have not been assigned to any clusters. At least three suggestions have been made in scientific literature, all of which have caveats: Unclustered points could be excluded from the evaluation, they could all be merged into one "garbage" cluster, or each unclustered point could form its own cluster for the purpose of evaluation [137]. The first option favors clustering algorithms that more aggressively remove difficult-to-cluster points, while the second biases the results towards recall and the third biases them towards precision.

A related problem is the interpretability of standard f-measure for the purpose of Relation Discovery. As [25] have previously argued, since the task is to identify potentially interesting relations in a corpus, overall precision may be considered more important than recall: High precision indicates that those clusters that have been identified are indeed valid relations. On the other hand, high recall can point to several distinct gold relations being incorrectly conflated into one cluster, making the results less helpful. Such considerations may mean that I may need to value precision higher than recall, and that unclustered points may not need to be penalized if this causes overall clustering results to be more helpful.

Chosen Approach Because of this, each set of experiments is conducted both quantitatively (i.e. measuring standard precision, recall and f-measure numbers) as well as qualitatively with regards to the above considerations.

Evaluation Measures Used I use **BCubed** for extrinsic clustering evaluation [138], an effective measure that satisfies the following essential criteria for measuring cluster quality [7]: *Cluster homogeneity*, which rewards clusterings with pure clusters. *Cluster completeness*, which promotes "same label, same cluster" policy. *Rag bag*, which rewards introducing a garbage cluster over polluting pure clusters. And *small cluster preservation*, which penalizes spreading data points of a rare label across various clusters.

General BCubed precision and recall are computed based on *multiplicity*, a measure of the minimum intersection between two data points o_i and o_j regarding their labels and cluster assignments. In our case this intersection contains 1 element at most, since we performed non-overlapping clustering. Depending on whether precision or recall is computed, multiplicity is normalized with the amount of shared cluster assignments or shared labels respectively:

$$multiplicity_{precision}(o_i, o_j) = \frac{min(|C(o_i) \cap C(o_j)|, |L(o_i) \cap L(o_j)|)}{|C(o_i) \cap C(o_j)|} \tag{3.2}$$

$$multiplicity_{recall}(o_i, o_j) = \frac{min(|C(o_i) \cap C(o_j)|, |L(o_i) \cap L(o_j)|)}{|L(o_i) \cap L(o_j)|} \tag{3.3}$$

[7] Cf. [139, Ch. 1, p. 6] for a more verbose elaboration on these quality criteria and BCubed in general.

Here $C(o_i)$ denotes the set of cluster assignments of a data point o_i given a clustering and $L(o_i)$ the set of labels for a given data point o_i according to ground truth. Precision and recall are then calculated by averaging multiplicity over all data points [8]:

$$Precision_{BCubed} = \frac{\sum_{i=1}^{n} \frac{\sum_{o_j : C(o_i) \cap C(o_j) \neq \emptyset} multiplicity_{precision}(o_i, o_j)}{\|\{o_j | C(o_i) \cap C(o_j) \neq \emptyset\}\|}}{n} \tag{3.4}$$

$$Recall_{BCubed} = \frac{\sum_{i=1}^{n} \frac{\sum_{o_j : L(o_i) \cap L(o_j) \neq \emptyset} multiplicity_{recall}(o_i, o_j)}{\|\{o_j | L(o_i) \cap L(o_j) \neq \emptyset\}\|}}{n} \tag{3.5}$$

In a final step, the F$_1$-score is calculated as the harmonic mean of $Precision_{BCubed}$ and $Recall_{BCubed}$.

3.3.2 Experiment 1: k-Means at Fixed k

In the first experiment, I run k-means on the four datasets with a fixed k that reflects the number of relations that are known to be contained in the datasets. So, for R10-ZIPFIAN and R10-ZIPFIAN, k is set to 10, while for R20-UNIFORM and R20-ZIPFIAN, k is set to 20. This first experiment is to examine the impact of different pattern generation methods on different datasets in the scenario that there exists a good assumption on how many distinct relations are contained in a dataset.

3.3.2.1 Quantitative Evaluation

R20 Datasets I first run all pattern generation methods on the two R20 datasets. The evaluation results are given in Table 3.5. I make three main observations:

1. Deep syntactic patterns consistently outperform shallow baselines. Both PROPOSED-ESP and PROPOSED-SUB outperform the shallow baselines. Of the three baselines, BASELINE-BOL is the best, reaching a precision of 0.55 and an f-measure of 0.3 on R20-ZIPFIAN, and a precision of 0.36 and an f-measure of 0.35 on R20-UNIFORM (highlighted bold in Table 3.5). Even without type restrictions, the proposed approaches outperform the best baseline, with PROPOSED-ESP reaching a precision of 0.63 and an f-measure of 0.39 on R20-ZIPFIAN, and a precision of 0.41 and an f-measure of 0.43 on R20-UNIFORM.

[8] Note that self-relation is not excluded. And that multiplicity is defined only when the two data points share at least 1 cluster assignment or label respectively.

			R20-Zipfian			R20-Uniform		
	Type Restrictions	pPMI	P	R	F1	P	R	F1
Baseline-TUR	None	no	0.39	0.19	0.26	0.21	0.23	0.22
		yes	0.5	0.19	0.27	0.15	0.4	0.22
Baseline-WAN	Ner	no	0.36	0.21	0.27	0.22	0.31	0.26
		yes	0.49	0.21	**0.3**	0.28	0.34	0.31
Baseline-BOL	None	no	0.47	0.22	0.3	0.24	0.25	0.25
		yes	**0.55**	0.21	**0.3**	**0.36**	0.34	**0.35**
Proposed-SUB	None	no	0.6	0.27	**0.37**	**0.39**	0.38	**0.39**
		yes	0.62	0.26	0.36	0.35	0.38	0.37
	Ner	no	0.57	0.29	0.39	0.41	0.46	0.43
		yes	0.51	0.28	0.36	0.39	0.47	0.42
	Phrasal-1	no	0.24	0.9	0.38	0.08	0.96	0.15
		yes	0.23	0.92	0.37	0.1	0.92	0.19
	Phrasal-5	no	0.52	0.36	0.43	0.42	0.63	0.5
		yes	0.5	0.44	0.47	0.42	0.57	0.48
	Phrasal-Full	no	0.57	0.33	0.42	0.38	0.55	0.45
		yes	0.58	0.33	0.42	0.37	0.51	0.43
	Yago	no	0.61	0.4	0.48	0.41	0.7	0.51
		yes	**0.67**	0.4	**0.5**	**0.48**	0.64	**0.55**
Proposed-ESP	None	no	0.63	0.24	0.35	**0.41**	0.44	**0.43**
		yes	0.63	0.28	**0.39**	0.38	0.43	0.41
	Ner	no	0.64	0.3	0.41	0.4	0.55	0.46
		yes	0.62	0.32	0.42	0.37	0.52	0.43
	Phrasal-1	no	0.3	0.77	0.43	0.13	0.92	0.22
		yes	0.28	0.82	0.42	0.12	0.92	0.22
	Phrasal-5	no	0.64	0.33	0.43	0.45	0.59	0.51
		yes	0.61	0.39	0.47	0.44	0.59	0.51
	Yago	no	**0.66**	0.39	**0.49**	0.49	0.69	**0.58**
		yes	0.63	0.36	0.46	**0.53**	0.64	**0.58**

Table 3.5: Evaluation of different pattern generation methods for the two **R20** datasets.

2. Fine-grained type restrictions greatly improve clustering quality. For the proposed approaches, I look into the effect of additionally using different type restrictions, indicated in the column "Type Restrictions" in Table 3.5. As can be seen, the greatest improvements are observed using fine-grained entity types derived from the Yago knowledge base: For Proposed-SUB on R20-Zipfian, f-measure improves from 0.37 to 0.5 when using Yago types. Similarly, on R20-Uniform, f-measure improves from 0.39 to 0.55. Similar improvements are observed for Proposed-ESP.

3. Phrasal clustering as type restrictions improve clustering quality. Focusing more specifically on the idea of using phrasal clusters as type restrictions, I also note significant improvements: Both the Phrasal-Full and Phrasal-5 setups increase precision and f-measure for all setups against the baselines of using no restrictions or only Named Entity types. The only difference is Phrasal-1 which decreases precision and f-measure against the baselines. This points to the necessity of using *overlapping* clustering when modeling type restrictions in an unsupervised way; if entities are assigned only to one cluster, crucial information seems to be lost.

			R10-Zipfian			R10-Zipfian		
	Type Restrictions	pPMI	P	R	F1	P	R	F1
Baseline-TUR	None	no	0.35	0.28	0.31	0.3	0.34	0.32
		yes	0.48	0.31	0.38	0.54	0.52	0.53
Baseline-WAN	Ner	no	0.35	0.26	0.3	0.3	0.37	0.33
		yes	0.27	0.54	0.36	0.44	0.52	0.47
Baseline-BOL	None	no	0.43	0.33	0.37	0.41	0.44	0.42
		yes	**0.53**	0.35	**0.42**	**0.48**	0.52	**0.5**
Proposed-SUB	None	no	0.54	0.36	0.43	**0.64**	0.67	**0.66**
		yes	0.57	0.38	0.46	0.61	0.63	0.62
	Ner	no	0.41	0.61	0.49	0.54	0.65	0.59
		yes	0.44	0.4	0.42	0.52	0.61	0.56
	Phrasal-1	no	0.21	0.98	0.35	0.11	0.99	0.2
		yes	0.22	0.95	0.36	0.14	0.94	0.24
	Phrasal-5	no	0.51	0.42	0.46	0.54	0.63	0.58
		yes	0.47	0.5	0.49	0.44	0.67	0.53
	Phrasal-Full	no	0.43	0.67	0.52	0.52	0.65	0.58
		yes	0.55	0.4	0.46	0.57	0.63	0.6
	Yago	no	0.52	0.54	0.53	0.54	0.68	0.6
		yes	0.51	0.63	0.56	0.51	0.72	0.6
Proposed-ESP	None	no	0.56	0.4	0.47	0.56	0.61	0.58
		yes	0.61	0.49	0.54	0.57	0.69	0.62
	Ner	no	0.54	0.42	0.48	0.54	0.6	0.57
		yes	0.5	0.42	0.45	0.6	0.64	0.62
	Phrasal-1	no	0.26	0.87	0.4	0.21	0.86	0.33
		yes	0.24	0.95	0.38	0.19	0.9	0.31
	Phrasal-5	no	0.55	0.4	0.47	0.55	0.77	0.64
		yes	0.54	0.43	0.48	0.49	0.69	0.57
	Yago	no	0.5	0.6	0.54	0.54	0.82	**0.65**
		yes	**0.56**	0.65	**0.6**	0.55	0.74	0.63

Table 3.6: Evaluation of different pattern generation methods for the two **R10** datasets.

R10 Datasets While these experiments affirm the initial hypotheses of using typed dependencies and fine-grained entity type restrictions, there remain a number of unanswered questions, such as the effects of pattern reweighting and dataset properties. I repeat the same set of experiments on the two R10 datasets and make a number of additional observations:

1. Impact of feature reweighting depends on the setup. I repeat each experiment once with reweighting features using pPMI ("pPMI" column set to "yes" in Tables 3.5 and 3.6) and without, simply using the co-occurrence counts as weights ("pPMI" column set to "no"). Looking through the experiments on all four datasets and with all different pattern generation methods, I note that pPMI sometimes improves and sometimes decreases performance. While it improves f-measure across the board for all baselines by as much as increasing from 0.32 to 0.52 for setup Baseline-TUR on R10-Zipfian, the picture for the proposed pattern generation approaches is mixed. For instance, for setup Proposed-SUB with restrictions None and Ner on R10-Zipfian, I observe slight decreases in overall f-measure. These observations point to feature reweighting being more effective for less informed pattern generation methods that generate more noise. Nevertheless, even for the proposed methods, I observe improvements in

None			
ID	**Example patterns**	**Example entity pairs**	**Label**
1	[Y] be son of [X] [Y] ’s father [X] [Y] born to [X]	*<Louis Bonaparte, Napoleon III>* *<Ron Paul, Rand Paul>* *<Tim Sweeney, Boston>*	ChildOf
2	[Y] performed by [X], video.n by [X] performing [Y], [Y] song.n written by [X],	*<ABBA, One of Us>* *<Wyclef Jean, Million Voices>* *<Haddaway, Life>*	Created
3	[Y] be [X] ’s album, listen free to [X] [Y], [X] [Y] released in,	*<ABBA, Ring Ring>* *<Vanessa Carlton, Big Yellow Taxi>* *<David Bowie, Tonight>*	Created
4	[Y] featuring [X] check out [Y] [X] [X] make [Y]	*<Vanilla Ice, Under Pressure>* *<Akin, Platform game>* *<Bobby Jindal, Republican Party>*	*Mixed*
Yago			
ID	**Example patterns**	**Example entity pairs**	**Label**
5	[Y:books] by [X:people] be book [X:people] author of [Y:books] [Y:books] book by [X:people]	*<Charles Dickens, American Notes>* *<Heinrich von Kleist, Penthesilea>* *<Michio Kaku, Parallel Worlds>*	Created
6	[X:companies] developed [Y:games] [Y:games] made by [X:companies] [X:companies] release [Y:games]	*<LucasArts, LOOM>* *<Neversoft, Guitar Hero>* *<Atari Games, Gauntlet>*	Created
Phrasal-Full			
ID	**Example patterns**	**Example entity pairs**	**Label**
7	[Y:296] written by [X:236] [Y:764] written by [X:662] [X:236] ’s novel [Y:624]	<Woody Allen, Without Feathers> <Jude Law, Rage> <Alex Garland, Sunshine>	Created

Table 3.7: 7 sample clusters found with setups None, Yago and Phrasal-Full. Each cluster is characterized by the top patterns in its centroid and represents one discovered relation. The entity pairs that make up the cluster are instances of discovered relations. Cluster 3, for example, represents the ChildOf relation which holds between two persons.

a majority of cases and only occasional slight decreases. I conclude that pPMI has an overall (albeit small) positive effect.

2. Better results with more data. A general observation is that the results are better across the board for all experiments on R10 against R20. This is perhaps unsurprising: Since all datasets have the same number of sentences, this means that there are more sentences per relation in the two R10 datasets. I therefore observe that the approach works better if more data is available per relation.

3.3.2.2 Qualitative Analysis

I manually inspect a sample of the discovered relations and patterns to gain insight into how the different setups affect the Relation Discovery capabilities of the Relation Discovery method. I illustrate these observations with a number of clusters shown in Table 3.7. I give examples of clusters for three setups, the proposed **PROPOSED-SUB** pattern extraction with NONE, YAGO and PHRASAL type restrictions respectively. I inspect results on dataset R20-ZIPFIAN. For each cluster, which represents one discovered relation, I list a small set of representative patterns and entity pairs, as well as the YAGO relation label of the majority of the entity pairs in the cluster. A majority label needs to be shared by at least 50% of all entity pairs in the cluster, otherwise no clear majority label is determined (this case is indicated by *Mixed* in column "Label").

Cluster **1** in Table 3.7, for example, is a cluster that represents a relation between a parent and its child, as is indicated by top patterns such as "`[Y] be son of [X]`" and "`[Y] 's father [X]`". Entity pairs in this cluster, such as <*Louis Bonaparte, Napoleon III*>, are instances of this relation. I find that this cluster corresponds closely to the CHILDOF relation from the YAGO knowledge base, indicating that this cluster is a positive example of Relation Discovery.

Large relations split up, small relations not found. The main observation when manually going through the data in all setups is that the relations with the most entity pairs in the dataset, namely CREATED and LOCATEDIN are split up into multiple clusters. The CREATED relation in YAGO for instance is a very broad relation between a person and any work of art that she created. However, this relation is never found within a single cluster but rather in multiple clusters with CREATED as majority label. Clusters **2** and **3** in Table 3.7 for instance share the same majority label, but are each characterized by different groups of patterns. This decreases recall, but does not impact precision in the evaluation.

On the other side of the spectrum I note that relations with few entity pairs are not found at all by the approach. Rather, they are conflated into larger, nonsensical clusters as well as a large garbage cluster. An example of this is cluster **4**, for which no majority label could be determined. It is characterized by a number of ambiguous patterns, such as "`[Y] featuring [X]`" and "`[X] make [Y]`" that could point to any number of distinct relations. A total of 5 out of 20 clusters with setup NONE are such clusters, indicating the negative impact of ambiguous and underspecified patterns.

Error reduction and finer granularity with YAGO and PHRASAL-FULL type restrictions. I also note that using YAGO and PHRASAL-FULL type restrictions significantly reduces ambiguities in patterns and leads to fewer clusters without majority labels. In fact, in both setups only 1 large cluster (the "garbage" cluster) remains without majority label. The large relations are also split along more meaningful lines. Focusing again on the CREATED relation, I note that it is split into various clusters, including clusters **5** and **6** which represent more fine-grained relations. Cluster **5** is a relation between a person and a book she has written. Cluster **6** is a relation between a company and a game it created. In the automatic evaluation, this split reduces recall, but it may be argued that such a distinction between two types of the broad CREATED relations is in fact desirable.

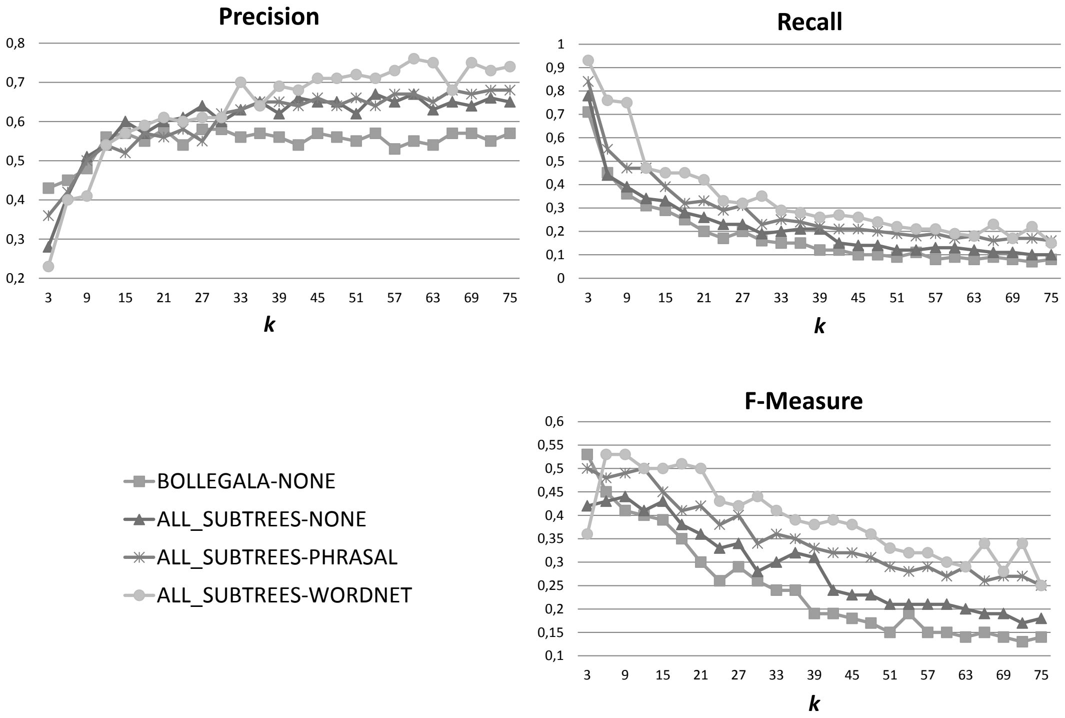


FIGURE 3.6: Clustering quality in terms of precision, recall and F$_1$-measure on the R20-ZIPFIAN dataset. The proposed pattern generation approaches all outperform BOL, the best baseline approach. Within the proposed approaches, YAGO entity type restrictions outperform the other setups.

Readability of patterns. Generally, I note that using named classes for type restrictions (NER and especially YAGO) result in more human readable patterns than their counterparts in the NONE and PHRASAL-FULL methods. Consider clusters **6** and **7**. Cluster **6** is easy to evaluate, as the top patterns are human readable. Cluster **7**, on the other hand, is characterized by patterns that contain cluster IDs. It is necessary to consult the entity pairs to determine that this cluster represents a relation between a person and a novel or screenplay she created.

3.3.3 Experiment 2: k-Means at Variable k

Given the results of the previous set of experiments, I now wish to determine the effects of setting higher k. Specifically, I would like to determine whether this allows me to control the granularity of discovered relations and whether this allows me to discover relations that are further down the long tail. In addition, I would like to see whether the proposed pattern generation method outperforms the baseline methods regardless of the parameterization of the clustering approach.

3.3.3.1 Quantitative Evaluation

I execute k-means for all k between 3 to 75 and measure precision, recall and f-measure as in the previous set of experiments. Here, I focus on R20-ZIPFIAN, the dataset expected to be the most like real data. The results of this experiment are shown in Figure 3.6. There are three plots shown here, one each for precision, recall and f-measure. The y-axis is the setting for k, while the x-axis is the respective evaluation measurement. For readability purposes, I only plot the results for the best baseline, and a few selected settings of the proposed approaches.

The general plot follows the expected lines: At low k, recall is highest since entity pairs are assigned to only a small number of clusters, increasing the chance that two entity pairs with the same label are indeed in the same cluster. This value drops as k increases. Precision values, on the other hand, are highest at high values for k, since this results in more clusters with fewer entity pairs, generally resulting in cluster of higher purity. The f-measure is the harmonic mean between the two. I make a number of observations from this experiment:

1. Proposed pattern generation methods outperforms baselines at any k. The most important observation is that at any setting for k, the proposed approach outperforms the baselines, pointing to the fact that an informed pattern generation method can significantly aid approaches for Relation Discovery. In addition, I note that the proposed method for modeling type restrictions using phrasal clustering also outperforms the baselines, even if YAGO types outperform this approach.

2. High f-measure at expected k. I also observe that the overall f-measure is highest for low values for k and comparatively high for the expected value of $k = 20$. While the best results are measured for k between 6 and 9, this is mostly due to the very high recall that naturally occurs at low k.

3.3.3.2 Qualitative Analysis

In the qualitative evaluation in experiment 1, I noted that a problem for Relation Discovery using clustering is that prominent relations tend to be distributed across numerous clusters while relations in the long tail are not discovered at all. Encouraged by the results of the parameter sweep evaluation which shows that the proposed pattern generation methods lead to higher precision and f-measure across all settings for k, I repeat a qualitative evaluation at higher k. Specifically, I set k to 60, i.e. three times the number of expected relations, which the experimental results show as the peak setting for precision for the R20-ZIPFIAN dataset at $k = 60$ with pattern generation PROPOSED-SUB and YAGO restrictions. I manually inspect the clusters and make a number of observations:

1. Long tail relations discovered at higher k. The first observation is that by setting higher k, the method finds clusters for relations that are missed at lower values for k. The YAGO relation ISAFFILIATEDTO for instance, is not identified at $k = 20$, but is found in two clusters at $k = 60$, one representing a relation between a person and the political party she is affiliated to, while the other represents a relation a person and her sport club.

2. Prominent relations split into even more clusters. However, this comes at the price that prominent relations are split into even more clusters than at lower k. For instance, the CREATED relation is the majority label of 26 clusters in this setup. Some of these splits are meaningful in the sense that they make a distinction between more granular relations of CREATED, such as PERSONCREATEDSONG, PERSONCREATEDALBUM, GROUPCREATEDALBUM and COMPANYCREATEDGAME. However, others seem to be redundant, with multiple clusters for instance centering around the relation GROUPCREATEDALBUM represented by different groups of correlating patterns.

These observations point to difficulties in practical applications for Relation Discovery which I discuss in the next Section.

3.4 Discussion and Summary

In this chapter, I investigated an unsupervised approach for Relation Discovery. I proposed pattern generation methods based on typed dependencies and fine-grained entity type restrictions. In particular, for the open domain, I presented a method that makes use of a Web-derived phrasal clustering of n-grams as type restrictions. It therefore does not require a pre-specified entity type hierarchy. In a series of experiments on four evaluation datasets, I observed that Relation Discovery using the proposed pattern generation method consistently outperforms a number of baselines.

However, the qualitative evaluation indicates that while prominent relations are reliably discovered, less prominent relations can only be found with parameterization that causes *1)* prominent relations to be fragmented across many clusters and *2)* a large number of clusters being returned by the approach. The main problem here is that ultimately, the “correct” granularity depends on the information need: For instance, the distinction between PERSONCREATEDALBUM and GROUPCREATEDALBUM may either be relevant or too fine-grained depending on the purpose for which the relations will be used. Even within the same dataset, the desired granularity may be different on a relation-by-relation basis. In the end, only human judgment can determine the relevance of relations and their granularity for a particular information need. Since the clustering approach however operates without regards for a specific information need, it is *undersupervised* (or "*ignorant of the questions*" in the words of this chapter's opening quote). This imposes limitations on how optimally a clustering approach can be designed and parameterized.

A related problem is that distributional evidence is sometimes, but not always meaningful. The evaluation has shown that the same relation may be expressed in different *groups* of patterns, causing the distributional evidence to indicate that the same relation should be split across multiple clusters. On the other hand, I note that some distinct relations correlate in the evidence. For instance, the BORNIN, LIVESIN and DIEDIN relations all strongly correlate since the same entity pairs are often persons who have been born, lived in and died in the same location. Similarly the GOVERNOROF and BORNIN strongly correlate since governors are nearly always born in the state they govern. Distributional evidence therefore directs the clustering approach to conflate distinct relations into one cluster. This again points to the conclusion that a discovery approach that is purely driven by correlations in the data has natural limitations.

I conclude that while distributional evidence is a valuable tool for Relation Discovery, especially with an informed pattern generation method, there needs to be a more direct way for a human to influence the discovery process; On the one hand, a user could supply domain or world knowledge to correct errors in which distributional evidence is misleading. On the other hand, a user could direct a discovery approach towards the relations and the granularity that she judges to be relevant. This is investigated in further detail in the next chapter.

4

Exploratory Relation Extraction

“’Incariol”’, Achamian finally said, “Why that name?” The Nonman’s stride did not falter. “Because I wander”. The Wizard breathed deep, knowing the time had come to plunge back into the fray. He squinted up at the figure. ”And ’Cleric’?” The Nonman’s pace slowed a fraction. A scowl furrowed his hairless brow. ”It is a tradition.. I think.. A tradition among the Siqû to take a Mannish name.”

– Excerpt from [140]

4.1 Overview

The research presented in the previous chapter found that at least some measure of user interactions are required to direct a Relation Discovery process towards a user-defined information need and to supplement distributional evidence with domain or world knowledge. Motivated by these observations, this chapter looks to rule-based Relation Extraction approaches which offer specific and direct control for users when creating extractors. I propose to combine aspects from Relation Discovery and rule-based RE in order to strike a balance between allowing more control on the part of the user, while preserving the cost-advantage of unsupervised approaches. I refer to the proposed approach as Exploratory Relation Extraction (ERE).

This section gives an overview over the challenges associated with rule-based RE, especially with regards to working with vague information needs and corpora of unknown content. In Section 4.1.2, I look at previous work in rule-based RE, Exploratory Search and preemptive Information Extraction. In Section 4.2, I propose the paradigm of Exploratory Relation Extraction and give an overview over the underlying principles, the proposed workflow and introduce SCHNÄPPER, a Web-based tool for executing this workflow. With SCHNÄPPER, I conduct an extensive user-study, the results of which are discussed in Section 4.3.

This chapter is mainly based on three previously published papers, namely [37] and [38], as well as to a lesser degree [39].

4.1.1 Problem Statement

Recent years have seen several trends in RE. One the one hand, there has been a renewed interest in manually created and maintained rule-based RE systems [67, 141]. Advantages of such systems include a better transparency and explainability of extraction rules, and the resulting maintainability and customizability of rule sets. Another trend in RE is to make increasing use of deep syntactic information in extractors [62], as dependency parsers become faster and more robust on irregular text [99]. The research presented in the previous chapter gives strong evidence that dependency tree subtrees and fine-grained type restrictions are a powerful abstraction layer for defining Relation Extraction patterns.

Bringing both trends together are recent works in the field of Open Information Extraction (OpenIE). Our OpenIE system KRAKEN [45] as well as the CLAUSIE [46] system both use a set of hand crafted rules on dependency trees to outperform previous classification based approaches. The latter system outperforms OLLIE [142], the machine learning based state-of-the art OpenIE system that uses dependency tree features. Not only does CLAUSIE report significant precision gains over OLLIE, but also finds 2.5 to 3.5 times more relations. These results indicate a strong potential for manually creating rule-based Relation Extraction systems using dependency trees. The higher level syntactic representation, I argue, may even facilitate rule writing. As was illustrated in Section 2.3.2, a higher linguistic abstraction means that much linguistic variation such as inserted expressions must not be specifically addressed. This would therefore enable the creation of more succinct RE rules, leading to better explainability and easier maintenance.

The problem: Practical limitations of rule-based RE. However, while such research makes a case for users to directly work with deep syntactic patterns to create extractors, a number of practical limitations exist:

Cost One of the main problems with rule-based approaches are the high costs associated with creating extractors and maintaining rule-sets. While I argue that defining patterns over dependency trees potentially reduces the complexity of the task, the flip side is that working with deep syntactic features requires rule-writers to have a background in computational linguistics. As such, manually creating extractors remains exclusive to trained specialists, and is considered to be both costly and time-consuming.

Relation Discovery Generally, rule-based RE systems require a careful upfront definition of extraction tasks before an extractor can be created. As noted in the introduction, practical scenarios are however often characterized by vague information needs and large corpora

> of unknown content [9]. In such scenarios, it is impractical to embark on creating extractors before relations of interest have been identified for which reasonable amounts of relation instances are assumed to be available in the corpus at hand. Generally, the high costs associated with creating rule-based extractors make such approaches unsuitable to Relation Discovery.

In this chapter, I investigate how the advantages of direct control offered by rule-based RE can be leveraged in the context of Relation Discovery (Chapter 3). Specifically, the question is how costs for developing rule-based extractors can be reduced to such a degree as to allow for exploratory analysis of unknown corpora at minimal effort, but maximal control.

4.1.2 Previous Work

The proposed approach to addressing these limitations draws inspiration from a number of lines of previous work:

Manual Rule-Based RE. First and foremost, this work is build on work on the field of manual, rule-based RE. This lines of approaches has been observed to be the predominantly preferred industry solution due to interpretability of extraction rules and easy adaption to changing domains [9, 67]. One of the major challenges associated with rule-based RE systems is the lack of tools to assist rule developers [9, 39, 67].

Recent work in the field of rule-based RE has investigated workflows and tooling to facilitate the creation of extractors. [15] presented a wizard-like approach to guide users in the process of building extractors. In [39], we investigated an example-driven workflow that allows even users who are unfamiliar with NLP to write extractors using lexico-syntactic patterns over dependency trees. Similarly, [17] created a toolkit for experts in a *domain* of interest, but not in NLP. Users create extractors for pre-defined entities and relations by seeding example instances in a semi-supervised fashion. [16] used a similar bootstrapping approach and created a tool for visualizing learned patterns for diagnostic purposes. Finally, [143] focus on reducing effort in a user-driven process by including elements from active learning and bootstrapping, but target their tool at NLP experts.

In contrast to the approach I propose in this chapter, these approaches are mostly intended for traditional RE in which relations of interest are specified in advance. I instead seek to enable an *exploratory* workflow in which relations of interest may be discovered through user interactions with available data at little effort.

Exploratory Search. The proposed approach is based on *Exploratory Search* (ES) [144, 145], an information seeking paradigm in the field of Information Retrieval. In ES, users begin an exploration process with an imprecise information need and progressively discover available information to address and sharpen it. This approach is commonly used for Web search engines, for example, when searching for Web sites to address an information need for which the user is initially unsure of how to formulate keyword search queries. Since ES is used for Information Retrieval, previous work has not applied this paradigm to tasks other than retrieving documents. I instead apply this paradigm to RE in order to allow users to explore an unknown corpus for structured, relational information of interest and create relation extractors in the process.

One of the challenges associated with the often desired capability of ES is the design of interactive interfaces to support users as they navigate through complex environments [144]. Similarly, one challenge this chapter investigates is how to create an intuitive workflow that allows non-experts in NLP to engage in relation exploration.

Preemptive and Open Information Extraction. Another source of inspiration for the proposed approach comes from Preemptive Information Extraction [27], as well as work in Open Information Extraction (OpenIE) [146] that builds on this idea. The idea of these approaches is to do one pass over a corpus and (pre-emptively) extract so-called *facts* from a corpus, which each consist of an entity pair and a *fact phrase* that in prose describes the relationship between the two entities. This idea is of interest, since human-readable fact phrases are a more intuitive abstraction for users than lexico-syntactic patterns. The price for this simplification however is that the distinction of what constitutes a fact is made in a one-pass fashion using syntactic rule-sets [46], classifiers [147] or both [148], dismissing all other text. If users only work with pre-extracted facts, they lose direct access to the corpus and can therefore only work with a reduced view of the data at hand.

Another crucial issue is the difference between facts and semantic relations as considered in this thesis, namely that facts do not address problems of synonymy and polysemy. While a relation can be expressed in text in many ways, such as "`[X] marry [Y]`" and "`[X] be married to [Y]`", facts simply use the words as they appear in the text as fact phrase. This results in distinct facts for semantically synonymous relations if expressed differently in text.

I draw inspiration from these approaches in the following ways: In the approach proposed in this chapter, I present lexico-syntactic patterns to users in a way reminiscent of OpenIE fact phrases in order to make working with patterns intuitive. I also conduct a one-pass extraction of patterns, but crucially do not make a distinction between fact and non-fact at extraction time. Rather, I preemptively extract all possible patterns (since every pattern may be useful to a user) and build a pair-pattern matrix-like data structure for the entire corpus. This allows users to select and group patterns into relation extractors. Users therefore work with a simplified representation, but have access to a far wider range of potential patterns than OpenIE approaches offer.

4.1.3 Contributions

I propose the novel paradigm of *Exploratory Relation Extraction* as a novel method for working with unknown corpora and vague information needs. I propose a process of *exploration* for relations of interest in available data that combines advantages from unsupervised Relation Discovery and rule-based Relation Extraction. I draw inspiration from the information seeking paradigm of Exploratory Search in which users start with a vaguely defined information need and - with a mix of look-up, browsing, analysis and exploration - progressively discover information available to address it and simultaneously concretize their information need. The proposed exploration process is intended to be usable even by novice users at minimal effort, and therefore addresses the challenges of rule-based IE systems outlined above.

In addition to proposing this novel paradigm, further contributions are:

Data-Guided Workflow I introduce a guided, interactive workflow aimed at allowing users to explore parsed text corpora for relations at minimal effort. Exploratory queries return matching relation instances and source sentences, as well as suggestions for further queries computed from the available data. By following a process of experimental querying and accepting or rejecting pattern suggestions, users identify relations of interest and group patterns into extractors. The goal is to make use of such data-guidance to facilitate exploration while giving as much explicit control to a user as possible.

Natural Language-Like Pattern Queries. I propose to use natural language-like queries that read like OpenIE fact phrases, but in fact match dependency tree subtrees and entity type restrictions. By displaying only the lexical portion of the patterns, and dismissing all deep syntactic information, I propose to simplify the execution of the proposed workflow to such a degree as to be usable even by non-NLP experts.

Evaluation on Large-Scale Data. I conduct two experiments on a large corpus of over 160 million sentences from the CLUEWEB09 to determine in how far non-experts can use ERE to discover and extract relations. The study indicates that with minimal preparation, novice users can execute ERE and build their own high-precision extractors. I discuss the results of the user study in detail, as well as strengths and weaknesses of my proposed approach.

4.2 Exploratory Relation Extraction

I first outline the proposed user-driven and gata-guided approach to relation exploration in Section 4.2.1 and give two example executions of the workflow to illustrate how the approach can be used to work with unknown corpora. In Section 4.2.2, I focus on the pattern abstraction layer and illustrate how I preemptively extract all subtrees in dependency trees from a given text corpus to prepare data-guidance. In Section 4.2.3, I then introduce SCHNÄPPER, a Web-based toolkit for executing this workflow.

4.2.1 Data-Guided Exploratory Workflow

4.2.1.1 Overview

A key component of the proposed approach is to provide data-guidance in the exploration process by computing suggestions for patterns from user input and enabling an interactive workflow that allows users to work with available data. Such guidance is needed for two reasons: First, though much effort is invested in human-readable extraction patterns, users may need support in formulating patterns and choosing entity type restrictions. This is especially the case when users are non-experts in the domain of interest and they strive to identify a range of appropriate patterns. Second, users may be uncertain of the information content of a given text corpus. By providing guidance through automatic pattern suggestions that reflect available information, the system can help users find patterns for their information need.

Initial query. The *initial query* is the first user interaction that launches the exploration process. The user supplies this initial query by providing a pattern that may be underspecified. Since patterns in this thesis consist of dependency tree subtrees and subject and object type restrictions, the initial query may specify any one or more of these three components. For instance, the entry point may be the pattern "`[X] write [Y]`", the object entity type BOOK, or both together. If more components of a pattern are specified in the initial query, a more focused exploration process is launched, while fewer components typically mean that a wider net is cast. With appropriate tooling, the formulation of this initial query can be guided through auto-complete options. For instance, by beginning to type a pattern that contains the word "write", a system can offer prominent patterns observed in the corpus that contain this word ("`[X] write [Y]`" or "`[X] write about [Y]`") for user selection.

Once an initial query is specified, the system retrieves a list of all matching entity pairs from the corpus. Using this list of entity pairs, the system also retrieves and counts all patterns and entity type restrictions observed for these entity pairs. These correlating patterns and type restrictions are then ordered by count. The underlying idea is to compute correlations using the same principle of distributional similarity as used in Relation Discovery (Chapter 3). However instead of computing the similarity of entity pairs using patterns, I compute the similarity of patterns using entity pairs. This returns three ordered lists of suggestions, one for each field: One list of suggested subject type restrictions, one list of subtree suggestions and one list of suggested object type restrictions. Tooling can again assist users in understanding these suggestions by providing example sentences in which pattern suggestions match.

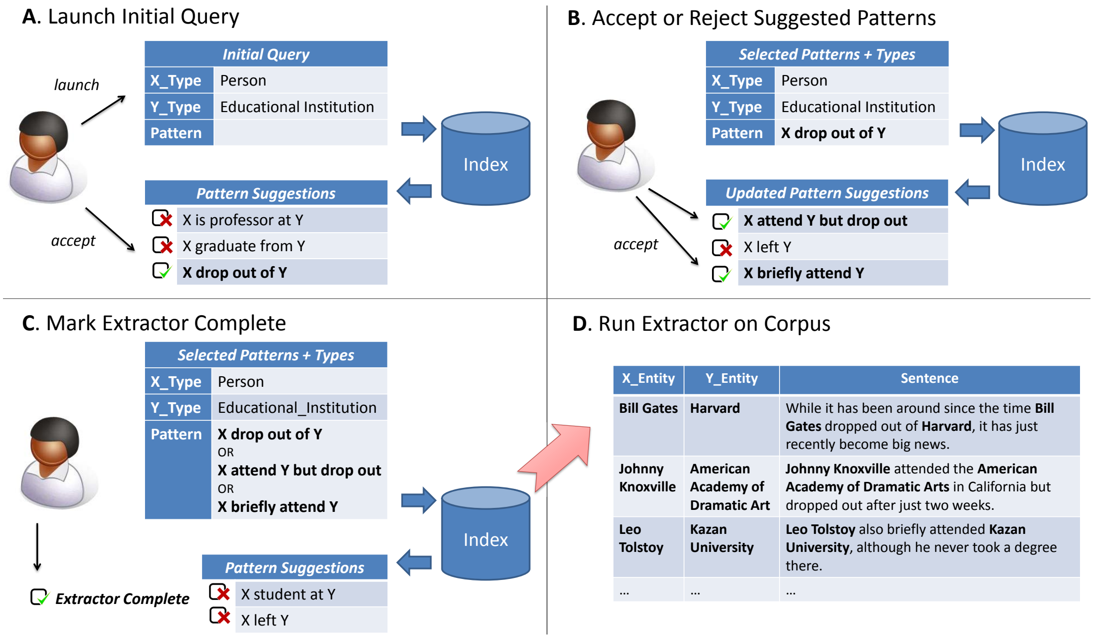
User interactions. After the initial query, the user is presented with a list of correlating pattern and entity type suggestions, from which the user can now pick and chose those suggestions that she believes to be relevant. Since each interaction (selection or de-selection of pattern components) affects the set of entity pairs the query matches, each interaction also updates the pattern suggestions. If more components are selected, the query will typically match a more narrow subset of entity pairs, which will therefore prompt more specific suggestions. The user therefore starts a process of selecting (and de-selecting) entity type restrictions and subtrees, thus refining the extractor while being guided by constantly updated pattern suggestions. The user continues this process until satisfied with the created extractor at which point it can be saved and the discovered relation instances downloaded.

In order to illustrate the proposed process, I give two example executions of such a workflow in the ensuing subsections.

4.2.1.2 Example 1: Vague Information Need

The first example is a use case in which an information need is described in prose, but for which exact extraction tasks yet need to be defined. The information need is find information on persons and their educational background. As per the scenario of this thesis, the corpus at hand is of unknown content, so at the onset of the exploration we are unsure of what type of relevant information may be found in the corpus. The goal is to use ERE to identify relations of interest and to build extractors for them. Refer to Figure 4.1 for an illustration of the first steps of this example.

Initial query (1). Knowing only that relations should hold between entities of type PERSON and entities of type EDUCATIONALINSTITUTION, the user issues an initial query with only these entity type restrictions, leaving the subtree field blank. This is illustrated in Figure 4.1 (A). A query is run against the data structure returns both a list of sentences that match the query (not illustrated in Figure 4.1), as well as a list of common patterns that hold between entities of such types, including "`[X] be professor at [Y]`", "`[X] study at [Y]`" and "`[X] drop out of [Y]`". By clicking on a pattern, the user retrieves entity pairs and sentences in which a pattern matches; For example, the user is informed that the pattern "`[X] study at [Y]`" finds the relation instance <*Bill Gates, Harvard University*> in the sentence "***Bill Gates** briefly studied at **Harvard University***.".


A. Launch Initial Query
launch
accept
Index
B. Accept or Reject Suggested Patterns
accept
Index
C. Mark Extractor Complete
Index
Extractor Complete
D. Run Extractor on Corpus

Initial Query	
X_Type	Person
Y_Type	Educational Institution
Pattern	

Pattern Suggestions
☒ X is professor at Y
☒ X graduate from Y
✓ **X drop out of Y**

Selected Patterns + Types	
X_Type	Person
Y_Type	Educational Institution
Pattern	**X drop out of Y**

Updated Pattern Suggestions
✓ **X attend Y but drop out**
☒ X left Y
✓ **X briefly attend Y**

Selected Patterns + Types	
X_Type	Person
Y_Type	Educational_Institution
Pattern	**X drop out of Y** OR **X attend Y but drop out** OR **X briefly attend Y**

Pattern Suggestions
☒ X student at Y
☒ X left Y

X_Entity	Y_Entity	Sentence
Bill Gates	**Harvard**	While it has been around since the time **Bill Gates** dropped out of **Harvard**, it has just recently become big news.
Johnny Knoxville	**American Academy of Dramatic Art**	**Johnny Knoxville** attended the **American Academy of Dramatic Arts** in California but dropped out after just two weeks.
Leo Tolstoy	**Kazan University**	**Leo Tolstoy** also briefly attended **Kazan University**, although he never took a degree there.
...	...	...

FIGURE 4.1: Illustration of the ERE example workflow discussed in Section 4.2.1.2.

Next to pattern suggestions, the initial query also returns suggestions for further subject and object type restrictions. For the purpose of readability, these suggestions are not depicted in Figure 4.1. In this example, I focus on the subtree suggestions.

Explore by reacting to suggestions until relation identified (2-4). Intrigued by the pattern "`[X] drop out from [Y]`", the user affirms this pattern and rejects all other suggestions. This causes a new query to be run against the parsed data, this time consisting of the entity restrictions as well as the pattern. As the query is now more concrete, the pattern suggestions are updated to reflect this new information. The user is presented with similar patterns such as "`[Y] dropout [X]`" and "`[X] attend [Y] but drop out`". This is illustrated in Figure 4.1 (B). The user repeats this, selecting or de-selecting patterns. At each interaction, suggestions are updated to reflect the current selection. When the user is satisfied with the identified relation, the selected set of patterns and restrictions is saved as an extractor (Figure 4.1 (C)) and executed against the entire text corpus (Figure 4.1 (D)). This returns lists of matching relation instances and sentences.

In this example, the user has thus started with a vague information need and identified a relation of interest in an unknown corpus, namely a relation for persons that attended an educational institution but did not graduate. The user can find additional relevant relations for this information need by repeating this workflow and interacting with the data at hand.

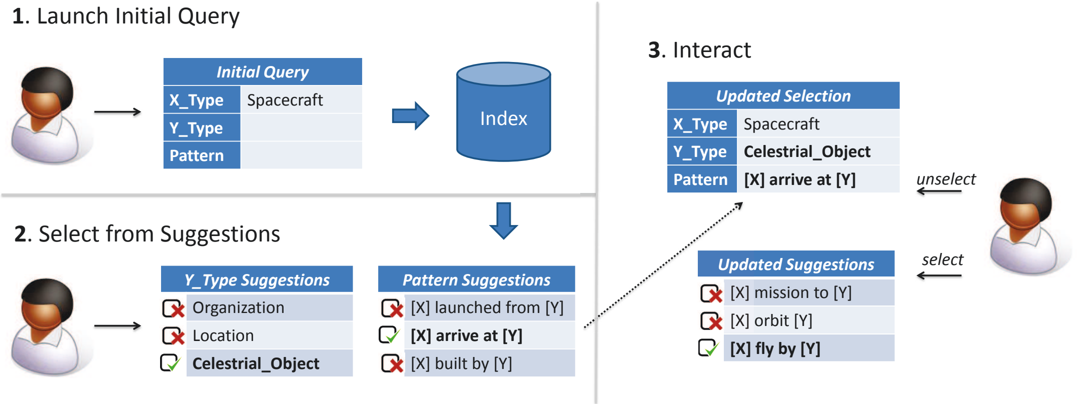


FIGURE 4.2: Illustration of the ERE example workflow discussed in Section 4.2.1.3.

4.2.1.3 Example 2: Exploratory Search for Relations

The second example is of a more exploratory nature, addressing a more vague information need than the previous example. Assume that our user is interested in relations that involve "spacecraft", but is unsure of what types of relations may be found for such entities in the given corpus. The goal is again to use ERE to identify relations of interest in this domain and to build extractors for discovered relations. The first steps of this example are illustrated in Figure 4.2.

Initial query (1). The user starts by issuing an initial query that is more strongly underspecified than in the previous example: The user sets only one of the entity type restriction fields to SPACECRAFT, leaving blank not only the *Pattern*, but also the object type restriction field. In effect, this means that the initial query is very broad, retrieving all sentences that contain at least one entity of type SPACECRAFT.

Explore by reacting to suggestions (2). After issuing the query, the system responds with both a list of sentences that match the query (not illustrated in Figure 4.2) and well as, more importantly, suggestions for patterns and object entity type restrictions that correlate with the user query. Since the initial query is more broad than in the previous example, the user has many possible options for directing the exploration process: For instance, by selecting the object type LOCATION and the pattern "`[X] launched from [Y]`", the user may direct the exploration process towards relations that indicate locations (cities, countries, sites) from which a spacecraft was launched. Similarly, by choosing ORGANIZATION as object type and "`[X] built by [Y]`" as pattern, the user may select organizations (contractors, space agencies) that constructed or designed spacecraft as the focus of interest.

In the example shown in Figure 4.2, the user instead selects the object type CELESTIALOBJECT and the pattern "`[X] arrive at [Y]`". This directs the search towards relations that indicate spacecraft missions to celestial objects.

Further user interactions (3). This user interaction updates both the query as well as the suggestions for patterns and restrictions. Now pattern suggestions are more specific to the previous selection; For instance, by selecting either the pattern "`[X] orbit [Y]`" or "`[X] fly by [Y]`", the user can specify relations for spacecraft that have achieved orbit around celestial objects, or have made flybys. This distinction between flybys and making orbit is more fine-grained than in the previous selection, but may nevertheless be relevant depending on the use case. By repeating this process of querying, inspecting results, selecting and de-selecting subtrees and restrictions, the user can find a set of interesting relations for her information need.

This example illustrates how even with a more exploratory information need and a more broad initial query, the proposed workflow can be applied to quickly narrow in on interesting relations and build appropriate extractors.

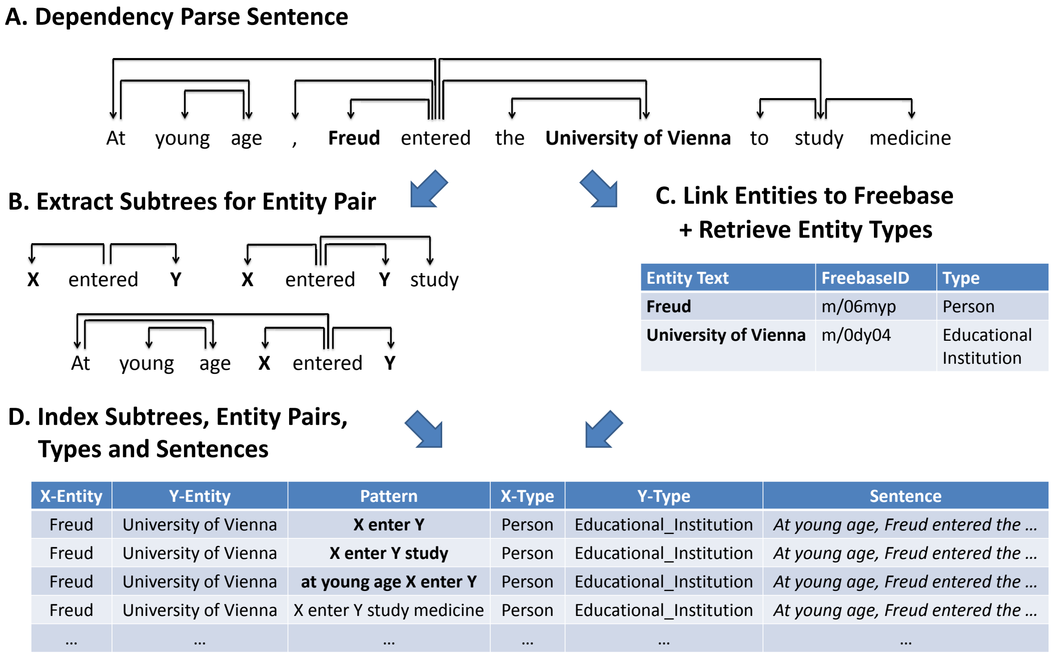
4.2.2 Exploratory Pattern Queries

4.2.2.1 Readability

A key point in lowering the entry barriers into the proposed user-driven approach is to make pattern queries and suggestion both easy to read and write for novice users. In Chapter 3, and particularly Section 3.3, I looked at a powerful abstraction layer for defining Relation Extraction patterns and analyzed different options with regards to human-readability. From this I draw the following insights with regards to human-readable ERE patterns:

Fine-grained type restrictions with human-readable descriptions The first is that fine-grained entity type restrictions can significantly contribute to extraction quality. While I investigated two options for modeling fine-grained type restrictions in Chapter 3, namely Wikipedia categories and phrasal cluster memberships, I noted that only the former has the advantage of human readability; Type restrictions such as UNIVERSITY, PERSON, PRODUCT or SPACECRAFT are intuitively interpretable while phrasal cluster IDs are not. For the purpose of readability, I therefore chose to use fine-grained NE types drawn from FREEBASE [84], arguably the largest publicly available knowledge base.

Lexicalized subtrees The second insight is that dependency tree subtrees are a powerful abstraction for defining patterns. However, as stated in the introduction of this chapter, one of the problems associated with deep syntactic features is that only persons with a background in computational linguistics are familiar enough with them as to be able to read and write queries in such a formalism. I therefore chose to render only a lexicalized form of the subtrees, i.e. the lexical forms of the words in the tree in the order they appear in the sentence. So, while these patterns are extracted from dependency parsed sentences, the user only interacts with their lexical representation, which often is human readable.


A. Dependency Parse Sentence
At young age , **Freud** entered the **University of Vienna** to study medicine
B. Extract Subtrees for Entity Pair
X entered Y
X entered Y study
At young age X entered Y
C. Link Entities to Freebase
+ Retrieve Entity Types
D. Index Subtrees, Entity Pairs,
Types and Sentences

Entity Text	FreebaseID	Type
Freud	m/06myp	Person
University of Vienna	m/0dy04	Educational Institution

X-Entity	Y-Entity	Pattern	X-Type	Y-Type	Sentence
Freud	University of Vienna	**X enter Y**	Person	Educational_Institution	*At young age, Freud entered the ...*
Freud	University of Vienna	**X enter Y study**	Person	Educational_Institution	*At young age, Freud entered the ...*
Freud	University of Vienna	**at young age X enter Y**	Person	Educational_Institution	*At young age, Freud entered the ...*
Freud	University of Vienna	X enter Y study medicine	Person	Educational_Institution	*At young age, Freud entered the ...*
...	...	...	...	...	...

FIGURE 4.3: Illustration of the subtree generation process. The system parses each sentence in a given document collection using a dependency parser and annotates all entities (**A**). It then generates all possible subtrees in the dependency tree that span pairs of annotated entities, three of which are illustrated in **(B)**, and link entities to their FREEBASE IDs to determine their entity types (**C**). The system then generates a lexical, lemmatized representation of these subtrees which is stored along with the entity pair, their entity types and sentence they are observed with **(D)**.

I argue that because patterns are lexicalized variants of dependency subtrees and entity type restrictions have human readable names, such queries are intuitive to users even without an NLP background.

4.2.2.2 Preemptive Extraction and Indexing

For ERE, I follow the idea of Preemptive Information Extraction [27] in which all possible relations for a given text corpus are preemptively generated in advance. Applied to ERE this means that PROPOSED-SUB pattern generation (see Chapter 3.2.1) is first performed on a corpus. In addition, the fine-grained NE types are determined for each entity in a corpus. All information is then stored in a data structure for fast retrieval and computation of pattern correlations. I illustrate the entire extraction and indexing process with an example sentence in Figure 4.3:

Dependency parse sentence (A) Each sentence in the corpus is first tagged with entities. Since I only consider binary relations, all sentences than contain fewer than 2 named entities are discarded at this point. The remainder of entity tagged sentences is then dependency parsed.

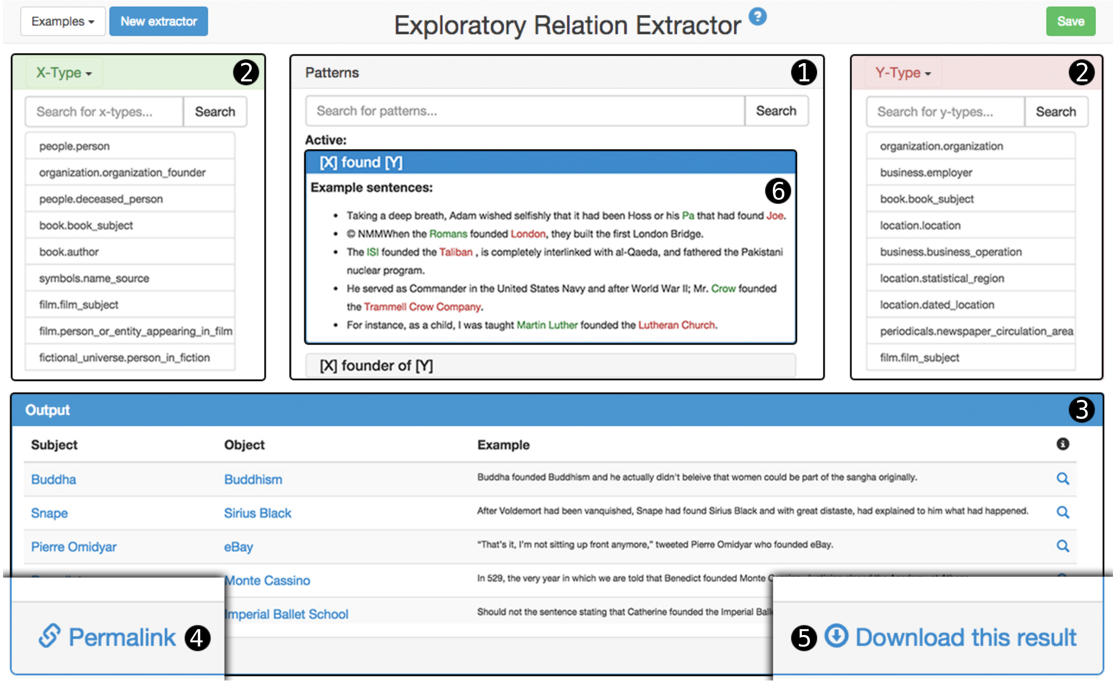
Extract subtrees for entity pair (B) For each entity pair in each sentence, the system extracts all connected subtrees (up to size s) that span both entities. The parameter s indicates the maximum number of nodes a subtree may consist of and may be set to limit the number of distinct subtrees extracted from a corpus. Practical experiments have shown that at s above 6, the pattern space becomes very sparse and computationally impractical. In the extracted subtrees, the entity tokens are replaced with the placeholders `[X]` and `[Y]`, where the former is the placeholder for the subject entity and the latter the placeholder for the object entity. The patterns are then lexicalized by lemmatizing the words and discarding information on typed dependencies, yielding flat pattern strings that are more human readable than subtrees. Therefore "`[X] enter [Y]`" and "`at young age [X] enter [Y]`" are two of the subtree patterns observed for the entity pair in the examples sentence.

Link entities to FREEBASE and retrieve entity types (C) In the next step, the systems needs to determine entity types for both entities. One option to accomplish this is through fine-grained NER, as offered for example by the FIGER system [112]. In the implementation, entities are instead linked to entries in the FREEBASE knowledge base, allowing the system to retrieve their fine grained entity types. This means that potentially more than one type may be retrieved for each entity.

Store in data structure (D) The system then indexes the information on lexicalized patterns, the entities they span and their types, as well as the sentences in which the patterns were found (Figure 4.3D).

When inspecting the lexicalized example subtrees in Figure 4.3 (B), we see how each of the subtrees conveys somewhat different semantics: For instance, the subtree pattern "`[X] enter [Y]`" may be found to denote a variety of relations depending on the entity types it is observed with. For example, it has a different meaning when observed between a PERSON and an EDUCATIONALINSTITUTION ("*At young age, **Freud** entered the **University of Vienna** to study medicine*") than when observed with two entities of type RIVER ("*The **Spoon River** enters the **Illinois River** opposite the town of Havana*"). The pattern "`[X] enter [Y] study`" more specifically points to a possible EDUCATEDAT relationship, while subtrees such as "`[X] enter [Y] to study medicine`" or "`at young age, [X] enter [Y]`" point to even more specific relationship types. By preemptively extracting all possible subtrees, I defer the decision of which is relevant to the individual user and information need.

The resulting data structure allows users to query for any combinations of patterns and entity type restrictions and quickly retrieve matching entity pairs and sentences from the index. Crucially, the data structure allows us to compute suggestions by agglomerating all patterns and entity type restrictions for all entity pairs found with a given query.



- people.person
- organization.organization_founder
- people.deceased_person
- book.book_subject
- book.author
- symbols.name_source
- film.film_subject
- film.person_or_entity_appearing_in_film
- fictional_universe.person_in_fiction

Active:

[X] found [Y]

Example sentences:

- Taking a deep breath, Adam wished selfishly that it had been Hoss or his Pa that had found Joe.
- © NMMWhen the Romans founded London, they built the first London Bridge.
- The ISI founded the Taliban , is completely interlinked with al-Qaeda, and fathered the Pakistani nuclear program.
- He served as Commander in the United States Navy and after World War II; Mr. Crow founded the Trammell Crow Company.
- For instance, as a child, I was taught Martin Luther founded the Lutheran Church.

[X] founder of [Y]

- organization.organization
- business.employer
- book.book_subject
- location.location
- business.business_operation
- location.statistical_region
- location.dated_location
- periodicals.newspaper_circulation_area
- film.film_subject

Subject	Object	Example
Buddha	Buddhism	Buddha founded Buddhism and he actually didn't beleive that women could be part of the sangha originally.
Snape	Sirius Black	After Voldemort had been vanquished, Snape had found Sirius Black and with great distaste, had explained to him what had happened.
Pierre Omidyar	eBay	"That's it, I'm not sitting up front anymore," tweeted Pierre Omidyar who founded eBay.
	Monte Cassino	In 529, the very year in which we are told that Benedict founded Monte
	Imperial Ballet School	Should not the sentence stating that Catherine founded the Imperial Ball

FIGURE 4.4: Screen capture of the SCHNÄPPER tool showing the *pattern panel (1)* with an activated pattern showing a list of example sentences (6), the *entity type restriction panels (2)* and the *result panel (3)*. The *permalink button (4)* and the *download button (5)* are located at the bottom.

4.2.3 The SCHNÄPPER Toolkit

For the purpose of evaluation and demonstration of the proposed approach, we[1] created the SCHNÄPPER Web toolkit for Exploratory Relation Extraction. It is briefly introduced in this subsection.

4.2.3.1 Web Interface

Since the tool is addressed at novice users, the user interface structures into four panels that fit onto one screen. The top half of the screen consists of three panels in which the user can select patterns and entity type restrictions. The bottom half of the screen is the result panel which displays a sample of extraction results for the currently selected patterns and entity type restrictions. See Figure 4.4 for the screen and a breakdown of the panels, which I explain in more detail in the following:

Pattern panel (1) Of the three panels in the upper half of the screen, the pattern panel assumes the center stage. Here, the user can enter keywords in the search field to find appropriate

[1] My student Thilo Michael implemented the Web UI [38].

patterns. If at least one user interaction has already been made (e.g. one pattern or type restriction selected), a list of pattern suggestions is presented in gray. Single clicking on a pattern suggestion gives a small number of example sentences and entity pairs for which this pattern holds (this is illustrated in field **(6)** in Figure 4.4). Double-clicking on a pattern adds it to the extractor; it is then highlighted blue and suggestions as well as the result panel are updated to reflect the selection. By double-clicking on a selected pattern, users may remove it again from the selection.

Entity type restriction panels (2) Extractors may also restrict lexico-syntactic patterns to only apply to entities of certain types. The top right and top left panels are used to define restrictions for the subject and object of a binary relation respectively. Here, users have a choice between three different ways of selecting entity type restrictions. The first and default option is to use FREEBASE entity types [84]. I.e. the user might select the subject of a relation to be only of the FREEBASE type SPACECRAFT, ORGANIZATION or CELESTIALOBJECT.

The user might also restrict a relation to one specific entity. For instance, by restricting the object of a BORNIN relation to be the country “Finland”, the extractor will only find persons born in Finland. Finally, the user can restrict entities to be only those found with a previously created extractor. Users can embed extractors in this way to find more complex relations. For instance, an extractor that finds “Persons born in Finland” may be used to restrict the subject entity of another extractor. The other extractor could then find a relation between “Persons born in Finland” and entities of type BUILDING, for example the relation “Buildings designed by persons from Finland”.

Similar to the pattern panel, double-clicking is used to select or unselect type restrictions. Upon each interaction, the suggestions as well as the result panel are updated to reflect the current selection.

Result panel (3) The lower half of the screen is the result panel which lists a set of entity pairs that are found with the presently selected patterns and restrictions. Each entity pair is displayed along with the sentence that matches the pattern. By clicking the magnifying glass symbol next to an entity pair, more details are shown, including the entity pair’s FREEBASE ids and a list of sentences that match the selected patterns.

Storing and exporting extractors After finishing building an extractor, users can export the setup as a JSON by clicking on the download button in the lower right corner of the screen (see field **(5)** in Figure 4.4). This exports the selected patterns and restrictions, together with a result list of entity pairs found with the extractor. In addition, users can generate a “permalink” by clicking the button in the lower left corner of the screen (see field **(4)** in Figure 4.4). This allows users to generate links to created extractors and share them electronically.

4.2.3.2 Example Usage

I now briefly give an example of using the tool. Assume a user is interested in a relation between persons and the companies that they have founded.

There are several entry points the user may choose from. For instance, the user might search for appropriate entity types in the *X_Type* and *Y_Type* panels. Another option is to start by looking for appropriate patterns. For this, the user can use the search box in the pattern panel **(1)** to search for the general term "found". This results in a list of patterns being displayed, which includes the pattern "`[X] found [Y]`". By single-clicking on it, the user can see a list of sentences that include this pattern. This is illustrated in field **(6)** in Figure 4.4.

The user activates the pattern by double-clicking it and then sees the output of the extractor in the result panel **(3)** as well as patterns and entity types that are suggested based on the current selection. Scanning through the result panel, the user finds that while many matching sentences do indeed express the desired relation (like "***Pierre Omidyar*** *founded* ***eBay***"), some others do not ("***Snape*** *found* ***Sirius Black***").

The tool also presents three sets of suggestions that the user can use to refine the patterns. For instance, for both *X_Type* and *Y_Type* a ranked list of suggestions highlighted gray appears **(2)**. As illustrated in Figure 4.4, it suggests PERSON as *X_Type* and ORGANIZATION as *Y_Type*. The user can affirm suggestions by double clicking on them. When selecting ORGANIZATION as *Y_Type*, the result panel is updated to reflect the most recent changes. Scanning through the results the user sees that the extraction quality has greatly improved as there are far fewer false positives in the list.

The user may now try to further improve the extractor by selecting more specific patterns. The tool suggests the pattern "`[X] be founder of [Y]`", which more accurately describes the relation the user wants to extract. Again by single-clicking on the suggestion, the user can see example sentences that match this pattern, as well as the selected entity type restrictions. Double-clicking on the pattern adds it to the extractor, which now consists of two patterns. With multiple patterns selected, the tool is now able to suggest patterns more accurately, offering patterns such as "`[Y] founded by [X]`", "`[X] start [Y]`" and "`[X] co-found [Y]`". By selecting them and implicitly rejecting those suggestions that do not reflect the desired relation (like the correlated patterns "`[X] president of [Y]`" or "`[X] CEO of [Y]`"), the user incrementally creates an extractor.

After multiple iterations of selecting suggested patterns and entity type restrictions the user is able to download the results of the extractor by using the download button (5) at the bottom of the page.

4.3 Evaluation

I seek to determine in how far the proposed approach addresses the problem of allowing users to openly explore an unknown corpus and create extractors at minimal effort. Since one of the central claims is that the approach is easily usable for persons even without a background in computational linguistics, I conduct a study in which I ask users to solve tasks with SCHNÄPPER, while giving them only a minimal amount of schooling into using the tool.

4.3.1 Experimental Setup

An evaluation of the proposed approach is challenging since it entails estimating both the *usability* of the approach as well as its capability for *exploration*. In this section, I give details on the dataset used for evaluation and the design of the user study, which includes the tasks presented to the users as well as their introduction to SCHNÄPPER.

4.3.1.1 Dataset

In order to ascertain the feasibility of using ERE on large corpora, I require large amounts of dependency parsed sentences annotated with named entities. I also require a large subset of these sentences to be annotated with gold standard relation labels in order to automatically evaluate user-created extractors.

I use a combination of three publicly available datasets to create the evaluation dataset: As source of text data, I use the English language portion of the well-known CLUEWEB09 [21] reference corpus, consisting of roughly 5 billion crawled Web pages. The preprocessing removes all HTML markup [149] and segments the resulting text into sentences. For entity detection and linking, I use an existing dataset released by Google Research, namely the FACC1 [150] resource. This resource is the result of a high quality named entity linking effort that was executed on the CLUEWEB09 corpus, linking over 6 billion entity mentions to their corresponding FREEBASE entries.

Using these two datasets, I identify over 160 million sentences in CLUEWEB09 that contain at least two entities that can be linked to FREEBASE. These sentences are tokenized, lemmatized, part-of-speech tagged and dependency parsed using the ClearNLP toolkit [151].

Silver Standard Relation Annotations. As sources for gold standard relation labels, I use the annotations from the "Relation Extraction Corpus" [152] a large, human-judged dataset of five relations about public figures on Wikipedia that was released by Google. Four of these relations involve FREEBASE entities, namely BORNIN, DIEDIN, EDUCATEDAT and GRADUATEDWITHDEGREE. In addition, I use the relation labels from FREEBASE with which I annotate sentences in the corpus using distant supervision. This results in roughly 5% of all 160 million sentences in the evaluation dataset to be annotated with silver standard relation labels.

4.3.1.2 User Preparation

The 10 users that participated in the study had no background in computational linguistics and Information Extraction, but all but one were students of computer science. A 3-page *tutorial document* was provided that illustrated the workflow for finding the RIVERTRIBUTARYOFRIVER relation. The document gave a step-by-step example of executing the workflow, similar to the description in Section 4.2.3.2. Users emulated the example workflow once and were permitted to ask questions during this time. After this typically around 10-minute period, users were no longer allowed to contact the study supervisor.

4.3.1.3 Evaluation Tasks

The goal was to evaluate whether novice users can use the proposed workflow for Relation Discovery and extraction. Accordingly, I defined two different tasks for the users, short descriptions of which were included in the tutorial document.

Extraction task. The first is an *extraction task* in which users were given four clearly defined semantic relations and were asked to create extractors for these relations. I used the four relations from the Relation Extraction corpus, namely BORNIN, DIEDIN, EDUCATEDAT and GRADUATEDWITHDEGREE since there were large amounts of silver standard annotations available for these relations. This allowed me to evaluate the first task quantitatively, by computing the quality of the user-created extractors in terms of standard precision, recall and f-measure metrics against silver data. Next to these measurements, I also recorded the time spent per user per extractor, in order to quantify the cost for creating extractors in terms of time.

Exploration task. The second task was an *exploration task* in which users were asked to identify relations for a more exploratory information need, namely identifying "interesting" relations that pertain to celebrities. What constitutes an "interesting" relation was left to the user. The goal with this task was to determine in how far the proposed approach could be used by novice users to openly explore a corpus for relations of interest. Due to the nature of this goal, no automatic evaluation could be performed; rather, I performed a qualitative evaluation in which participants

	EducatedAt					GraduatedWithDegree				
	#Inst	P	R	#Pat	Time	#Inst	P	R	#Pat	Time
User 1	**58,611**	0.99	0.2	**51**	12 min	17,698	1.0	0.27	34	17 min
User 2	48,782	0.99	**0.31**	34	15 min	12,180	1.0	0.27	27	14 min
User 3	25,435	0.88	0.12	12	8 min	**54,371**	0.93	0.53	24	8 min
User 4	33,095	0.99	0.23	25	12 min	7,196	1.0	0.22	9	10 min
User 5	47,668	0.76	0.16	29	13 min	34,942	1.0	0.48	3	5 min
User 6	20,356	0.99	0.15	18	14 min	10,290	1.0	0.25	12	14 min
User 7	22,889	0.62	0.01	8	4 min	37,119	0.71	0.6	19	4 min
User 8	31,412	0.98	0.19	13	15 min	1,251	0.46	0.04	10	14 min
User 9	14,169	0.99	0.1	6	8 min	13,104	0.6	0.17	13	12 min
User 10	29,289	0.99	0.19	17	15 min	35	1.0	0.02	4	20 min
Average	33,171	0.92	0.17	21	11.6 min	18,819	0.87	0.29	16	11.8 min

	BornIn					DiedIn				
	#Inst	P	R	#Pat	Time	#Inst	P	R	#Pat	Time
User 1	**158,222**	0.7	0.26	18	9 min	**25,779**	0.7	0.14	32	9 min
User 2	72,888	0.79	0.21	23	17 min	13,582	0.86	0.13	12	12 min
User 3	89,825	0.84	0.22	21	7 min	15,849	0.86	0.13	12	7 min
User 4	66,899	0.81	0.21	19	14 min	13,542	0.86	0.13	11	8 min
User 5	65,213	0.82	0.19	19	15 min	21,105	0.85	0.13	10	9 min
User 6	131,275	0.83	0.25	16	13 min	14,423	0.85	0.13	8	9 min
User 7	7,851	0.85	0.03	5	4 min	15,980	0.85	0.14	17	4 min
User 8	52,927	0.82	0.17	10	15 min	25,090	0.74	0.14	8	14 min
User 9	56,724	0.84	0.18	10	12 min	15,728	0.85	0.14	8	9 min
User 10	58,347	0.94	0.22	10	15 min	14,112	0.86	0.13	8	10 min
Average	76,017	0.82	0.19	15	12.1 min	33,171	0.82	0.13	13	9.1 min

Table 4.1: Evaluation results for the 4 well-defined relations in the extraction task.

were interviewed after the task was complete and asked a series of questions. I also collected and manually inspected their extractors.

4.3.2 Experiment 1: Extraction Task

4.3.2.1 Quantitative Evaluation

The 10 users created 4 extractors each (one for each relation), which was evaluated in terms of precision, recall and f-measure. However, even with relatively large sources of annotations, only roughly 5% of entity pairs in the 160 million sentences have a known Freebase relation. I therefore compute these measurements only for labeled entity pairs, and separately list the absolute number of extracted relation instances. A detailed overview over the results is given in Table 4.1.

I make a number of observations from these results:

Large amounts of relation instances at high precision. As Table 4.1 indicates, many users were able to create extractors that find very large amounts of instances (over 100.000 instances

False Positives		
Class	Count	Example Sentence
FB Mismatch	95	**Washington** died in his home in **Vermont** on December 14, 1799.
Type Error	82	[..] the scene where **Boromir** is killed in The **Fellowship of the Ring**.
FB Incomplete	14	[..] on December 27, **Dorr** died in **Providence**, in his native Rhode Island.
Other	9	[..] **Rascal Flatts** die in het schijfcd album omvatten **Feels Like Today**.

False Negatives		
Class	Count	Example Sentence
Common	87	**Klein** holds a **Bachelor of Arts**.
Long Tail	79	**Roger Blandford** is a native of England and took his **BA**, MA and [..].
Other	34	[..], 1974; **MS**, 1976; PhD, University of Pierre and **Marie Curie**, 1982.

TABLE 4.2: Analysis of 200 false positives and 200 false negatives to determine error classes for precision and recall loss. Each error class is listed with an example sentence. Main reasons for false positives included a mismatch in granularity between extraction results and annotations, wrongly specified types by the users or cases in which instances were found that were not in FREEBASE. Main reasons for false negatives were mostly patterns that users failed so select, either common patterns, or more rare patterns from the long tail.

in some cases) at high precision in an average time of 9 to 12 minutes, while recall values tend to be lower. This tendency to favor precision at the cost of recall has been observed in previous works on rule-based RE [153]. Nevertheless, the results raise the question of why recall numbers are comparatively low with this approach.

Users selected different number of patterns. I note differences from user to user, especially with regards to the number of found instances (**#INST**), the number of selected patterns (**#PAT**) and the time spent per relation. In Table 4.1, users are ordered by the total number of patterns they selected. User 1 selected the most patterns overall and found the most instances for the BORNIN, DIEDIN and EDUCATEDAT relations (highlighted bold). User 10 both spent the most time overall while selecting the fewest patterns. User 7 spent the least amount of time overall. This raises the question of why some users selected more patterns than others.

4.3.2.2 Qualitative Analysis

In order to gain insight into the questions raised by the quantitative evaluation, namely low recall and the differing user behavior, I first analyzed precision and recall in greater detail by manually evaluating a sample of 200 false positives and 200 false negatives by hand to discover the reasons for precision and recall loss. I determine principal error classes, which are listed with examples in Table 4.2.

Mismatch between gold standard and results. As Table 4.2 shows, false positives are most commonly due to inconsistencies between extraction results and the gold standard annotations concerning the level of granularity of a relation instance. For example, I found BORNIN and DIEDIN relation instances that indicated a person's place of birth or death at lower or higher granularity than FREEBASE records. An example of this is given in Table 4.2 for George Washington's place of death; the extractor finds the less granular <*Washington, Vermont*>, while the gold standard expects <*Washington, Mount Vernon*>[2]. While different from the gold standard, such instances are not false, which suggests that actual precision may be higher than the measured values indicate.

Missed patterns and entity types. The most common causes of recall loss are patterns that users failed to select. In Table 4.2, I distinguish between "common" patterns that were found by at least one user and "long tail" patterns that were found by none. While I did not expect a user-driven approach to identify long tail patterns, there was the question of why some users failed to find more common patterns. Similarly, the second most common cause of precision loss are entity type restrictions that users failed to correctly select, again against expectation. When interviewed, users named two main reasons for this:

1. **Halting problem** One of the main problems encountered by the users was the "halting problem", i.e. the question of when to stop adding patterns to an extractor. For some relations, such as BORNIN, users already found thousands of relation instances after selecting the first pattern, which caused two problems; First, they were unsure of the quality of the selected pattern(s), as they were unable to manually check thousands of relation instances for their validity. Second, they were unsure if more patterns were even needed if the first few already found such amounts of relation instances. Accordingly, some users selected many patterns (up to 51 distinct patterns for the EDUCATEDAT relation by user 1), while others selected only a few.

2. **Difficulties concerning entity types** Another main difficulty related to the precise meaning of FREEBASE entity types; For instance, there are several location types, such as LOCATION.LOCATION, LOCATION.DATED_LOCATION and LOCATION.STATISTICAL_REGION, which users found to be confusing, a problem that was compounded by occasional entity linking errors. Many users expressed the desire to specify custom entity types as restrictions in order to have a similar level of control here as over the choice of patterns.

4.3.3 Experiment 2: Exploration Task

I also asked users to explore the corpus for a vaguely defined information need, namely for relations that pertain to "celebrities", as well as one arbitrary relation. Users spent widely varying amounts of time (between 5 and 50 minutes) on this task due to differences in motivation, as

[2] "Mount Vernon" was George Washington's estate in the state of Vermont.

RELATION	EXAMPLE PATTERNS	EXAMPLE INSTANCES
CELEBRITYDIVORCE *(Divorces between celebrities)*	`[X] and [Y] divorce`, `[X] divorce [Y]`,	<Nicole Kidman, Tom Cruise> <Federline, Spears>
CELEBRITYDRIVESCAR *(Cars that celebrities drive)*	`[X] drives [Y]`, `[X] 's car [Y]`,	<Arnold Schwarzenegger, H1> <DiCaprio, Toyota Prius>
CONTESTEDBITHPLACE *(Speculative birthplaces of persons)*	`if [X] born in [Y]`, "`whether [X] born in [Y]`",	<Barack Obama, Kenya> <Barack Obama, Nigeria>

TABLE 4.3: Examples for relations discovered in the exploration task. CELEBRITYDIVORCE represents a commonly discovered relation, while CELEBRITYDRIVESCAR represents a relation that is presently not part of Freebase. CONTESTEDBITHPLACE is an example of a relation that utilizes closed-world words in patterns.

some users had interpreted the search for "interesting" relations as a challenge. For each relation, users provided a short description. I inspected the extractors qualitatively and interviewed the users.

4.3.3.1 Qualitative Analysis

Approach more suited to exploration than extraction. I found that users generally favored the exploration over the extraction tasks as here the search could be directed to more fine-granular and specialized relations. Issues that were observed for the extraction tasks, most notably the "halting problem", were not not encountered in the exploration tasks, as here users could decide the information need for themselves and select patterns accordingly.

Some relations not in Freebase. While the most common types of relations found for entities of type CELEBRITY regarded different types of romantic involvements with other celebrities such as marriages and divorces, some relations were identified that are not found in FREEBASE. This included a relation that connects a celebrity to the sports team they support or the car they drive. Examples for this are given in Table 4.3. This indicates a potential for using ERE to identify new relations for addition to existing knowledge bases.

Closed-class words can be relevant. Interestingly, one user also worked with patterns that involved closed-class word classes, such as "if" and "whether". Table 4.3 shows an example of a relation that indicates the speculative birthplaces of persons using such words. This gives an example of one of the core differences between ERE patterns and OpenIE facts, namely that the decision of what constitutes a relation and what does not is made by the user, while an OpenIE system will pre-filter facts that do not conform to certain patterns.

4.4 Discussion and Summary

In this chapter, I proposed Exploratory Relation Extraction as a novel method of exploring text corpora of uncertain content for relations of interest given an imprecise information need. I presented and evaluated a user-driven and data-guided incremental exploration workflow that enables non-expert users to identify relations and create high precision extractors with minimal effort.

The evaluation showed that users were generally able to start exploring the corpus using the proposed workflow immediately after the brief introduction. Users stated the natural language-like representation of patterns to be intuitively readable, although for some it required a trial and error process to understand how patterns matched entities in sentences. Even with no background in NLP, users created high precision extractors in a matter of minutes and were able to explore a corpus for relations of interest.

Future work building on ERE may explore a number of directions:

1. **Opt-in complexity** While the workflow stressed low entry barriers and exploratory search, some users wished to understand in greater detail how entity types are determined and whether this could be influenced. This indicates the need for adding options in future work that give more experienced users more technical information (and control) on dependency trees and FREEBASE types. Such "opt-in" complexity would allow us to hold on to the intuitive nature of the workflow for beginner users, while allowing experienced users more control in order to build higher quality extractors. Future work could investigate how entity-level extractors could be built using a similar approach, and what other types of features may be beneficial to the exploration and extraction process.

2. **Application to specific domains** Another interesting future direction would be to apply ERE to specific domains for which large corpora are available, for example the biomedical domain. A key point of interest would be whether a user-guided exploration process would uncover useful semantic relations that have so far been underexplored.

3. **Increasing recall** One of the main issues noted with ERE is that recall tends to be low since users typically select only a small number of patterns for a given relation. While this is less problematic for exploration, where the goal is only to identify interesting relations, this is more difficult for actual extraction tasks. A number of approaches to address this limitation are possible: For instance, once users have identified a relation of interest, distant supervision could be used for all instances returned by the prototypical extractor. Another possibility would be to shift user interactions towards an active learning-style learning process in which highly correlating patterns are semi-automatically added to an extractor, limiting user interactions only to contentious patterns.

With regards to the goals of this thesis, I conclude that the proposed method is a potent tool for working with vague information needs and unknown corpora. In the next chapter, I turn to the issue of multilingual data that has not been addressed so far.

5

Multilingual Semantic Role Labeling

He shook his head and cast his eyes heavenward, a mock gesture meant to tell Sorweel that he simply teased. Expressions, it seemed, all spoke in the same language.

– Excerpt from [154]

5.1 Overview

The previous two chapters investigated unsupervised and user-driven methods for working with corpora of unknown content but focused only on English-language text. This chapter looks more closely at the challenge of handling text in different languages. To lower the costs of developing multilingual extractors, I propose a method for multilingual Semantic Role Labeling that parses multilingual text into a shallow semantic, language-neutral abstraction. This both allows us to hide language-specific elements from the data scientist and enables extractors to work across many different languages at no additional cost. This chapter focuses on automatically generating appropriate resources that can be used to train such multilingual SRL. For this, I propose an *annotation projection* approach which I execute to generate PropBank-like SRL resources for 7 distinct languages from different language groups, namely Arabic, Chinese, French, German, Hindi, Russian and Spanish. An extensive evaluation of the generated resources indicates that the proposed method outperforms earlier annotation project works and that it is capable of generating medium to high quality resources for multilingual SRL.

This section gives an overview of the motivation behind multilingual SRL, the challenges of automatically generating resources to train appropriate parsers and previous work. It also gives an overview over the proposed approach and its contributions. Section 5.2 then illustrates the proposed two-step approach in detail, motivating and evaluating each step separately. Section 5.3

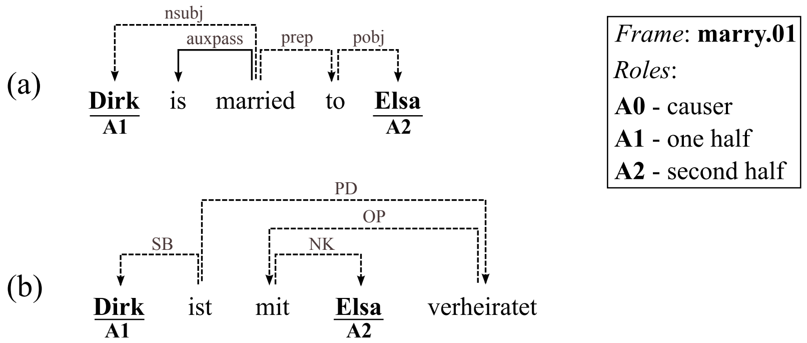


FIGURE 5.1: A simple English sentence (a) and its literal translation in German (b). The dependency parses given in arcs above the sentences are structurally different. On the other hand, the proposed multilingual Semantic Role labels are stable across both languages.

uses the approach to generate SRL resources for a set of languages and conducts an extensive evaluation of these resources. Finally, Section 5.4 discusses the results and their impact.

This chapter is mainly based on a previously published full paper [40].

5.1.1 Problem Statement

Multilingual text poses challenges for Relation Extraction since languages differ fundamentally on lexical, morphological and syntactic levels, affecting the pattern extraction step of the approaches that have been proposed in this thesis so far. For instance, the pattern "`[X] is married to [Y]`" can only match English-language text. Not only are lexical items such as "*married*" expressed differently in other languages, but also are syntactic constructions different from language to language. For instance, some languages such as German and Russian have a case system to mark nous according to their syntactic function, allowing a much greater freedom of word order than English [155]. Lexico-syntactic realization may vary greatly between languages: The same relation may be expressed by a verb, a noun or an expression in different languages, modifiers may be adjective or aspectual verb constructions and so forth [156].

Refer to Figure 5.1 for an example of a simple sentence in English and its literal German translation: The word order differs, as do the structure and the labels of the dependency parses. Because of such differences, it is necessary to create *separate extractors* on *separate feature sets* for each language [157, 158]. However, with regards to the goal of exploring multilingual data for relations of interest, treating each language separately is impractical, since it not only multiplies the effort but also requires users to be proficient in a range of different natural languages.

Crosslingual SRL as interlingual representation for RE Since users work with a human-readable abstraction for defining patterns, the question with regards to the goals of this thesis is whether an abstraction can be identified that is stable across different human languages. The concept of such a "language-neutral" or *interlingual* representation of semantics has been widely studied in the field of Statistical Machine Translation (SMT) [159–162]. However, it was found conceptionally to be too difficult to create an adequate representation: In order to translate text between different languages, an interlingual representation would need to be able to express the superset of fine-grained semantics of all languages [161].

I argue that since Relation Extraction is a form of shallow text analysis in which certain types of factual, structured information are extracted from text, an adequate interlingua for this task would not need to be as complex as for SMT. Rather, it would only need to capture coarse-grained, shallow semantics. For this, Semantic Role Labeling offers the basis of a potential solution. Since SRL is is the task of automatically labeling predicates and arguments in a sentence with shallow semantic labels, it is a representation that is more stable across syntactically different sentences. Furthermore, previous work has observed that semantic frames tend to be relatively stable even across different languages [31, 156, 163, 164]. However, SRL currently only has language-specific frame models, meaning that SRL will parse different languages into different shallow semantic representations. In addition, even for language-specific SRL, few comprehensive resources are available for other languages than English [30–33].

My goal is to enable multilingual SRL as the basis for multilingual RE. Refer to the SRL annotation in Figure 5.1 for an example of the proposed solution. Both the English sentence and its German translation use the same SRL annotation: The constituent "*Dirk*" is in both sentences marked as **A1** of the frame "MARRY.01". The constituent "*Elsa*" is always marked as **A2**. The annotation is therefore stable across languages.

The problem: Lack of training data for multilingual SRL In order to enable such multilingual Semantic Role Labeling I require an approach that automatically generates appropriate, multilingual training data. Such data needs to be generated for a range of languages in sufficient quantity (e.g. covering a broad range of shallow semantics and frame evoking elements) and quality (so that statistical SRL systems can be trained using this data). Furthermore, the multilingual resource needs to use identical frame labels across all languages.

I propose an approach to generate such data based on earlier work in *annotation projection*, which I introduce in the next section.

5.1.2 Previous Work

Annotation Projection of Semantic Labels As a cost-effective alternative to manual annotation, previous work has investigated the *direct projection* of semantic labels from a resource rich

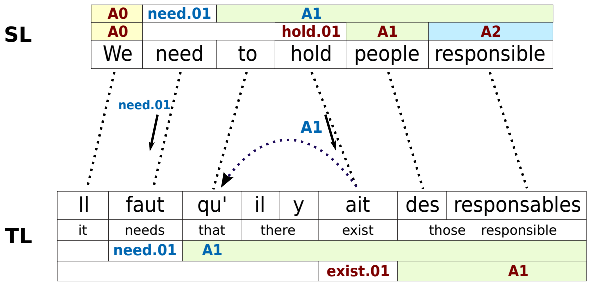


FIGURE 5.2: Pair of parallel sentences from French$_{\text{gold}}$ with word alignments (dotted lines), SRL labels for the English sentence, and gold SRL labels for the French sentence. Only two of the seven English SRL labels should be projected here.

language (English) to a resource poor target language (TL) in parallel corpora [109, 163]. The underlying assumption is that original and translated sentences in parallel corpora are semantically broadly equivalent. Hence, if English sentences of a parallel corpus are automatically labeled using an SRL system, these labels can be projected onto aligned words in the TL corpus, thereby automatically labeling the TL corpus with semantic labels. This way, PROPBANK-like resources can automatically be created that enable the training of statistical SRL systems for new TLs.

Consider the sentence pair in Figure 5.2, which consists of an English sentence from the Europarl corpus and its French translation. The English side is automatically labeled using SRL, the labels are given above the sentence. All English words are aligned to their French translations, as given in dotted lines. Since the French sentence is a translation of the English sentence, at least some of these semantic labels can be projected onto the French sentence.

Translation shifts. However, as noted in previous work [109, 163], aligned sentences in parallel corpora often exhibit issues such as *translation shifts* that go against this assumption. For example, in Figure 5.2, the English sentence "*I need to **hold** people responsible*" is translated into a French sentence that literally reads as "*There need to **exist** those responsible*". Hence, the predicate label of the English word "*hold*" should not be projected onto the French verb, which has a different meaning. As Figure 5.2 illustrates, this means that only a subset of all SL labels can be directly projected. This issue of translation shift is a major practical limitation to an annotation projection approach for semantic labels.

While initial studies considered the projection of FRAMENET labels [165, 166], more recent work focuses on PROPBANK due to its broader coverage and better availability of resources for English [167]. [109] scaled up direct projection of English PROPBANK labels to a French corpus and manually created a gold standard resource of 1,000 sentences to evaluate the French results. To address issues of translation shift, [168] presented an approach that aggregates information at the corpus level resulting in a significantly better SRL corpus for French. However, this approach has several practical limitations: (1) it does not consider the problem of argument identification of SRL systems, treating arguments as already given; (2) it generates rules for the argument classification step preferably from manually annotated data; (3) it has been demonstrated for a single language (French), and was not applied to any other language.

The proposed method is based on this previous work, but presents a novel approach to addressing translation shifts It generates resources to train an SRL system for both predicate and argument labels in a completely automatic fashion. In another important contrast to previous work which was only applied and evaluated on a single language, I apply the approach to generate PROPBANKS for 7 languages and conduct experiments over all languages.

Annotation Projection for Other Tasks Annotation projection has also been used to create NLP resources for other tasks. The idea was introduced in the context of learning a POS taggers, chunkers, NE taggers and morphological analyzers for languages such as French, Chinese, Czech and Spanish [169]. Other work has considered annotation projection for creating deep syntactic parsers [170, 171]. In order to address issues of translation shifts, [172] use token and type constraints to guide learning in cross-lingual POS tagging. Our proposed method draws inspiration from this approach.

Multilingual RE The single predominant method for multilingual RE is to create separate extractors for each language of interest. While research is scant on this topic, a few approaches have been discussed to address the issue of costs in multilingual settings: [20, 173] discuss using seed-based approaches in which, similar to a bootstrapping or distant supervision approach, instances are seeded in multilingual text to acquire patterns for different languages. [20] propose to translate all data into English and run existing English-language extractors on the translated text. [174] propose to project relation labels across parallel text to bootstrap extractors for other languages. However, none of these approaches are applicable to multilingual Relation Discovery since there are no shared feature representations and relations need to be pre-specified.

5.1.3 Contributions

I propose a method to generate multilingual SRL resources based on English PROPBANK labels. The proposed method operates in two stages to address the issue of translation shifts and is outlined in Figure 5.3: Given a parallel corpus in which the source language (SL) side is automatically labeled with PROPBANK labels and the TL side is syntactically parsed, the approach applies a *filtered projection* approach that allows the projection only of high-confidence SL labels. This results in a TL corpus with low recall but high precision. In the second stage, the approach repeatedly samples a subset of complete TL sentences and trains a classifier to

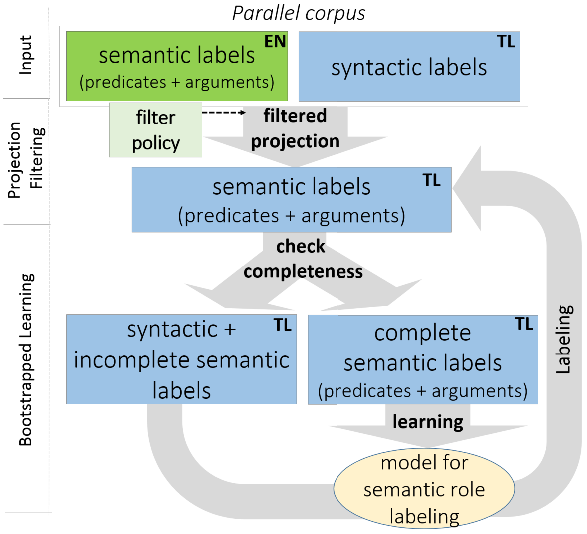

Parallel corpus
Input
semantic labels (predicates + arguments) EN
syntactic labels TL
Projection Filtering
filter policy
filtered projection
semantic labels (predicates + arguments) TL
check completeness
Bootstrapped Learning
syntactic + incomplete semantic labels TL
complete semantic labels (predicates + arguments) TL
learning
model for semantic role labeling
Labeling

FIGURE 5.3: Overview of the proposed two-stage approach for projecting English (EN) Semantic Role labels onto a TL corpus.

iteratively add new labels, significantly increasing the recall in the TL corpus while retaining the improvement in precision.

In more detail, the contributions are:

Filtered projection I conduct a detailed analysis of errors that occur in direct projection. Based on this analysis, I propose a set of filtering policies directed at detecting translation shifts. These policies are used to block low-confidence projections in the first step of the proposed approach, so that high precision, but low recall TL annotations are produced.

Bootstrap learning approach In order to address the low recall of the first step, I propose a bootstrapping approach in which TL SRL is repeatedly trained using a subset of sentences sampled from the intermediary TL corpus. At each iteration of the approach, the TL is supplemented with additional high precision labels. This approach is used to increase TL recall at small costs to precision.

Comprehensive evaluation I report on a comprehensive evaluation of the effectiveness and generalizability of the proposed approach over 7 different language pairs. I also investigate the impact of additional factors to the approach and discuss possible future work based on this analysis. Using the proposed approach, I generate PROPBANKS for all seven languages and release them to the research community.

5.2 Generating Multilingual SRL Resources

As outlined in Figure 5.3, the proposed approach consists of two stages (see Figure 5.3): The first step, filtered annotation projection, is explained in detail in Section 5.2.1. The second step, bootstrap learning, builds on the high precision output of this step and supplements additional labels to increase recall. It is described in detail in Section 5.2.2.

5.2.1 Stage 1: Filtered Annotation Projection

Stage 1 of the proposed approach is designed to create a TL corpus with high precision semantic labels. It is based on direct annotation projection [109] which transfers semantic labels from SL sentences to TL sentences according to word alignments. Formally, for each pair of sentences s_{SL} and s_{TL} in the parallel corpus, the word alignment produces alignment pairs $(w_{\text{SL},i}, w_{\text{TL},i'})$, where $w_{\text{SL},i}$ and $w_{\text{TL},i'}$ are words from s_{SL} and s_{TL} respectively. Under direct projection, if $l_{\text{SL},i}$ is a predicate label for $w_{\text{SL},i}$ and $(w_{\text{SL},i}, w_{\text{TL},i'})$ is an alignment pair, then $l_{\text{SL},i}$ is transferred to $w_{\text{TL},i'}$; If $l_{\text{SL},j}$ is a predicate-argument label for $(w_{\text{SL},i}, w_{\text{SL},j})$, and $(w_{\text{SL},i}, w_{\text{TL},i'})$ and $(w_{\text{SL},j}, w_{\text{TL},j'})$ are alignment pairs, then $l_{\text{SL},j}$ is transferred to $(w_{\text{TL},i'}, w_{\text{TL},j'})$, as illustrated below.

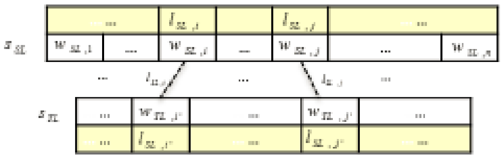


FIGURE 5.4: Illustration of direct projection.

Filtered Projection To address errors stemming from issues such as translation shifts, I propose *filtered projection* focused specifically on improving the precision of projected labels. Specifically, for a pair of sentences s_{SL} and s_{TL} in the parallel corpus, I retain the semantic label $l_{\text{SL},i}$ projected from $w_{\text{SL},i}$ onto $w_{\text{TL},i'}$ if and only if it satisfies the filtering policies. This results in a target corpus containing fewer labels but of higher precision compared to that obtained via direct projection.

I begin by analyzing typical errors in direct projection (Section 5.2.1.1). Based on these results, I design a set of filters to handle such errors (Section 5.2.1.2), and experimentally evaluate their effectiveness (Section 5.2.1.3).

5.2.1.1 Analysis of Direct Projection

I execute direct projection of English SRL to French SRL on a parallel corpus for which gold French SRL labels are available. This allows me to evaluate in how far projected labels are correct and what the principal classes of errors are. In more detail, I use the following experimental setup:

Data For experiments in this section and Section 5.2.2, I used the gold data set compiled by [109], referred to as $\texttt{French}_{\text{gold}}$. It consists of 1,000 sentence-pairs from the English-French `Europarl` corpus [22] with French sentences manually labeled with predicate and argument labels from the English PROPBANK.

Evaluation In line with previous work [175], I count synonymous predicate labels sharing the same VERBNET [176] class as true positives.[1] In addition, I exclude modal verbs from the evaluation due to inconsistent annotation.

Source Language SRL Throughout the rest of the chapter, I use CLEARNLP [96], a state-of-the-art SRL system, to produce semantic labels for English text.

The experimental results for direct projection are given in Table 5.3 (row labeld *Direct*). I observe that projection labels have both low precision and low recall and proceed to inspect the results to find error classes.

Analysis of False Negatives The low recall of direct projection is not surprising; most semantic labels in the French sentences do not appear in the corresponding English sentences at all. Specifically, among 1,741 predicate labels in the French sentences, only 778 exist in the corresponding English sentences, imposing a 45% upper bound on the recall for projected predicates. Similarly, of the 5,061 argument labels in the French sentences, only 1,757 exist in the corresponding English sentences, resulting in a 35% upper bound on recall for arguments.[2]

Analysis of False Positives While the recall produced by direct projection is close to the theoretical upper bound, the precision is far from the theoretical upper bound of 100%. To understand causes of false positives, I examine a random sample of 200 false positives, of which 100 are incorrect predicate labels, and 100 are incorrect argument labels belonging to correctly projected predicates. Table 5.1 and 5.2 show the detailed breakdown of errors for predicates and arguments, respectively. I first analyze the most common types of errors and discuss the residual errors later in Section 5.2.1.3.

[1] For instance, the French verb *sembler* may be correctly labeled as either of the synonyms: *seem.01* or *appear.02*.

[2] This upper bound is different from the one reported in [109] which corresponds to the inter-annotator agreement over manual annotation of 100 sentences.

ERROR CLASS	NUMBER
Translation Shift: Predicate Mismatch	37
Translation Shift: Verb→Non-verb	36
No English Equivalent	8
Gold Data Errors	6
SRL Errors	5
Verb (near-)Synonyms	4
Light Verb Construction	3
Alignment Errors	1
Total	100

TABLE 5.1: Breakdown of error classes in **predicate** projection.

ERROR CLASS	NUMBER
Non-Argument Head	33
SRL Errors	31
No English Equivalent	12
Gold Data Errors	11
Translation Shift: Argument Function	6
Parsing Errors	4
Alignment Errors	3
Total	100

TABLE 5.2: Breakdown of error classes in **argument** projection.

- **Translation Shift: Predicate Mismatch** The most common predicate errors (37%) are translation shifts in which an English predicate is aligned to a French verb with a different meaning. Figure 5.2 illustrates such a translation shift: label HOLD.01 of English verb "*hold*" is wrongly projected onto the French verb "*ait*", which is labeled as EXIST.01 in $\texttt{French}_{\text{gold}}$.

- **Translation Shift: Verb→Non-Verb** is another common predicate error (36%). English verbs may be aligned with TL words other than verbs, which is often indicative of translation shifts. For instance, in the following sentence pair

s_{SL}	I	know	what	happened
s_{FR}	On	connaît	la	suite
	We	know	the	result

  the English verb *happen* is aligned to the French noun "*suite*", causing it to be wrongly projected with the English predicate label HAPPEN.01.

- **Non-Argument Head** The most common argument error (33%) is caused by the projection of argument labels onto words other than the syntactic head of a target verb's argument. For example, in Figure 5.2 the label **A1** on the English "*hold*" is wrongly transferred to the French "*ait*", which is not the syntactic head of the complement.

5.2.1.2 Filtering Policies

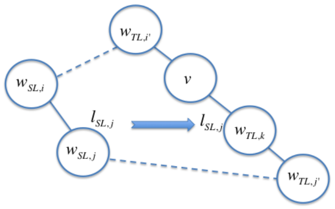
Based on the preceding error analysis, I consider the following filters to remove the most common types of false positives:

Verb Filter (VF) targets Verb→Non-Verb translation shift errors [109]. Formally, if direct projection transfers predicate label $l_{\text{SL},i}$ from $w_{\text{SL},i}$ onto $w_{\text{TL},i'}$, retain $l_{\text{SL},i}$ only if both $w_{\text{SL},i}$ and $w_{\text{TL},i'}$ are verbs.

Translation Filter (TF) handles both Predicate Mismatch and Verb→Non-Verb translation shift errors. It makes use of a translation dictionary and allows projection only if the TL verb is a valid translation of the SL verb. In addition, in order to ensure consistent predicate labels throughout the TL corpus, if a SL verb has several possible synonymous translations, it allows projection only for the most commonly observed translation.

Formally, for an aligned pair $(w_{\text{SL},i}, w_{\text{TL},i'})$ where $w_{\text{SL},i}$ has predicate label $l_{\text{SL},i}$, if $(w_{\text{SL},i}, w_{\text{TL},i'})$ is not a verb to verb translation from SL to TL, assign no label to $w_{\text{TL},i'}$. Otherwise, split the set of SL translations of $w_{\text{TL},i'}$ into synonym sets $S_1, S_2, \ldots$; For each k, let W^k be the subset of S_k most commonly aligned with $w_{\text{TL},i'}$; If $w_{\text{SL},i}$ is in one of these W^k, assign label $l_{\text{SL},i}$ to $w_{\text{TL},i'}$; Otherwise assign no label to $w_{\text{TL},i'}$.

Reattachment Heuristic (RH) targets non-argument head errors that occur if a TL argument is not the direct child of a verb in the dependency parse tree of its sentence.[3] Assume direct projection transfers the predicate-argument label $l_{\text{SL},j}$ from ($w_{\text{SL},i}$, $w_{\text{SL},j}$) onto ($w_{\text{TL},i'}$, $w_{\text{TL},j'}$). Find the immediate ancestor verb of $w_{\text{TL},j'}$ in the dependency parse tree. Denote as $w_{\text{TL},k}$ its child that is an ancestor of $w_{\text{TL},j'}$. Assign the label $l_{\text{SL},j}$ to ($w_{\text{TL},i'}$, $w_{\text{TL},k}$) instead of ($w_{\text{TL},i'}$, $w_{\text{TL},j'}$). An illustration is below:



RH ensures that labels are always attached to the syntactic heads of their respective arguments, as determined by the dependency tree. An example of such reattachment is illustrated in Figure 5.2 (curved arrow on TL sentence).

[3] In [165], a similar filtering method is defined over constituent-based trees to reduce the set of viable nodes for argument labels to all nodes that are not a child of some ancestor of the predicate.

	PREDICATE			ARGUMENT		
PROJECTION	P	R	F1	P	R	F1
Direct	0.45	0.4	0.43	0.43	0.31	0.36
VF	0.59	0.4	0.48	0.53	0.31	0.39
TF	**0.88**	0.36	0.51	0.58	0.17	0.27
VF+RH	0.59	0.4	0.48	0.68	0.35	0.46
TF+RH	**0.88**	0.36	0.51	**0.75**	0.2	0.31
Upper Bound	1	0.45	0.62	1	0.35	0.51

TABLE 5.3: Quality of predicate and argument labels for different projection methods on $\texttt{French}_{\text{gold}}$, including upper bound.

5.2.1.3 Effectiveness of Filtering Policies

I now present an initial validation on the effectiveness of the aforementioned filters by evaluating their contribution to annotation projection quality for $\texttt{French}_{\text{gold}}$, as summarized in Table 5.3.

VF reduces the number of wrongly projected predicate labels, resulting in an increase of predicate precision to 59% (↑14 pp), without impact to recall. As a side effect, argument precision also increases to 53% (↑10 pp), since, if a predicate label cannot be projected, none of its arguments can be projected.

TF [4] reduces the number of wrongly projected predicate labels even more significantly, increasing predicate precision to 88% (↑43 pp), at a small cost to recall. Again, argument precision increases as a side effect. However, as expected, argument recall decreases significantly (↓14 pp, to 17%), as many arguments can no longer be projected.

RH targets argument labels directly (unlike VF and TF), significantly increasing argument precision and slightly increasing argument recall.

In summary, initial experiments confirm that the proposed filters are effective in improving precision of projected labels at a small cost in recall. In fact, TF+RH results in nearly 100% improvement in predicate and argument labels precision with a much smaller drop in recall.

Residual Errors Filtered projection removes the most common errors that were discussed in Section 5.2.1.1. Most of the remaining errors come from the following sources:

SRL Errors The most common residual errors in the remaining projected labels, especially for argument labels, are caused by mistakes made by the English SRL system. Any wrong

[4] In all experiments in this chapter, I derived the translation dictionaries from the WIKTIONARY project and used VERBNET and WORDNET to find SL synonym groups.

label it assigns to an English sentence may be projected onto the TL sentence, resulting in false positives.

No English Equivalent A small number of errors occur due to French particularities that do not exist in English. Such errors include certain French verbs for which no appropriate English PropBank labels exists, and French-specific syntactic particularities.[5]

Gold Data Errors The evaluation so far relies on $\texttt{French}_{\text{gold}}$ as ground truth. Unfortunately, $\texttt{French}_{\text{gold}}$ does contain a small number of errors (e.g. missing argument labels). As a result, some correctly projected labels are being mistaken as false positives, causing a drop in both precision and recall. I therefore expect the true precision and recall of the approach to be somewhat higher than the estimate based on $\texttt{French}_{\text{gold}}$.

5.2.2 Stage 2: Bootstrapped Training of SRL

Since low-confidence projections are filtered out in stage 1 of the proposed method, the TL corpus at this point suffers from low recall. Stage 2 addresses this issue by iteratively supplementing labels using SRL trained over a high quality subset of the TL corpus.

Relabeling The proposed method draws inspiration from the idea of relabeling [109]. In relabeling, an SRL system is trained over a TL corpus that was produced with direct projection. This system is then used to relabel the corpus, effectively overwriting the projected labels with potentially less noisy predicted labels.

I first present an analysis on relabeling in concert with the proposed filters in Section 5.2.2.1. Based on this, I formulate a bootstrap algorithm in Section 5.2.2.2. I conduct an initial evaluation the bootstrapping approach over $\texttt{French}_{\text{gold}}$ in Section 5.2.2.1 and estimate parameters that are used in the experimental evaluation of Section 5.3.

5.2.2.1 Analysis of Relabeling Approach

I use the same experimental setup as in Section 5.2.1, and produce a labeled French corpus for each filtered annotation method. I then train an off-the-shelf SRL system [28] on each generated corpus and use it to relabel the corpus.

I measure precision and recall of each resulting TL corpus against $\texttt{French}_{\text{gold}}$ (see Table **??**). Across all experiments, relabeling consistently improves recall over projection. The results also show how different factors affect the performance of relabeling.

[5] French negations, for instance, are split into a particle and a connegative. In the annotation scheme used in $\texttt{French}_{\text{gold}}$, particles and connegatives are labeled differently.

Supplement vs. Overwrite Projected Labels The labels produced by the trained SRL can be used to either *overwrite* projected labels as in [109], or to *supplement* them (supplying labels only for words w/o projected labels). Whether to overwrite or supplement depends on whether labels produced by the trained SRL are of higher quality than the projected labels. I find that while predicted labels are of higher precision than directly projected labels, they are of lower precision than labels post filtered projection. Therefore, for filtered projection, it makes more sense to allow predicted labels to only *supplement* projected labels.

Impact of Sampling Method I am further interested in learning the impact of sampling the data on the quality of relabeling. For the best filter found earlier (TF+RH), I compare SRL trained on the entire data set (full data) with SRL trained only on the subset of completely annotated sentences (comp. sent.), where completeness is defined as:

Definition 1. A direct component of a labeled sentence s_{TL} is either a verb in s_{TL} or a syntactic dependent of a verb. Then s_{TL} is ***k*-complete** if s_{TL} contains equal to or fewer than k unlabeled direct components. 0-complete is abbreviated as **complete**.

I observe that for TF+RH, when new labels supplement projected labels, relabeling over complete sentences results in better recall at slightly reduced precision, while including incomplete sentences into the training data reduces recall, but improves precision. While this finding may seem counterintuitive, it can be explained by how statistical SRL works. A densely labeled training data (such as comp. sent.) usually results in an SRL that generates densely labeled sentences, resulting in better recall but poorer precision. On the other hand, training data that is sparsely labeled results in an SRL that weighs the option of not assigning a label with higher probability, resulting in better precision and poorer recall. In short, one can control the tradeoff between precision and recall of SRL output by manipulating the completeness of the training data.

5.2.2.2 Bootstrap Learning

Building on the observation that one can sample data in such a way as to either favor precision or recall, I propose a bootstrapping algorithm to train an SRL iteratively over k*-complete* subsets of the data which are supplemented by high precision labels produced from previous iteration. The detailed algorithm is depicted in Algorithm 1.

Algorithm 1.

```
Require: Corpus $C_{\text{TL}}$ with initial set of labels $L_{\text{TL}}$, and resampling threshold function
    $k(i)$;
    for $i = 1$ to $\infty$ do
        Let $k_i = k(i)$;
        Let ${C_{\text{TL}}}^{\text{comp}} = \{w \in C_{\text{TL}} : w \in s_{\text{TL}}, s_{\text{TL}}$is $k_i$-complete$\}$;
        Let ${L_{\text{TL}}}^{\text{comp}}$ be subset of $L_{\text{TL}}$ appearing on ${C_{\text{TL}}}^{\text{comp}}$;
        Train an SRL on $({C_{\text{TL}}}^{\text{comp}}, {L_{\text{TL}}}^{\text{comp}})$;
        Use the SRL to produce label set ${L_{\text{TL}}}^{\text{new}}$ on $C_{\text{TL}}$;
        Let ${C_{\text{TL}}}^{\text{no.lab}} = \{w \in C_{\text{TL}} : w$ not labelled by $L_{\text{TL}}\}$;
        Let ${L_{\text{TL}}}^{\text{suppl}}$ be subset of ${L_{\text{TL}}}^{\text{new}}$ appearing on ${C_{\text{TL}}}^{\text{no.lab}}$;
        if ${L_{\text{TL}}}^{\text{suppl}} = \emptyset$ then
            Return the SRL;
        end if
        Let $L_{\text{TL}} = L_{\text{TL}} \cup {L_{\text{TL}}}^{\text{suppl}}$;
    end for
end
```

Resampling Threshold The goal is to use bootstrap learning to improve recall without sacrificing too much precision.

Proposition 1. Under any resampling threshold, the set of labels L_{TL} increases monotonically in each iteration of Algorithm 1.

Since Proposition 1 guarantees the increase of the set of labels, I need to select a resampling function to favor precision while improving recall. Specifically, I use the formula $k(i) = \max(k_0 - i, 0)$, where k_0 is sufficiently large. Since the precision of labels generated by the SRL is lower than the precision of labels obtained from filtered projection, the precision of the training data is expected to decrease with the increase in recall. Therefore, starting with a high k seeks to ensure high precision labels are added to the training data in the first iterations. Decreasing k in each iteration seeks to ensure that resampling is done in an increasingly restrictive way to ensure that only high-quality annotated sentences are added to the training data, thus maintaining a high confidence in the learned SRL model.

5.2.2.3 Effectiveness of Bootstrapping

I experimentally evaluate the effectiveness of this model with $k_0 = 9$.[6] As shown in Table **??**, bootstrapping outperforms relabeling, producing labels with best overall quality in terms of F_1 measure and recall for both predicates and arguments, with a relatively small cost in precision.

While Algorithm 1 guarantees the increase of recall (Proposition 1), it provides no such guarantee on precision. Therefore, it is important to experimentally decide an early termination cutoff before the SRL gets overtrained. To do so, I evaluated the performance of the bootstrapping algorithm at each iteration (Figure 5.5). I observe that for the first 3 iterations, F_1-measure for both predicates and arguments rises due to large increase in recall which offsets the smaller drop in precision. Then F_1-measure remains stable, with recall rising and precision falling slightly at each iteration until convergence. To optimize precision and avoid overtraining, I set an iteration

[6] I experimentally determined that setting k_0 to larger values had little impact on the final results .

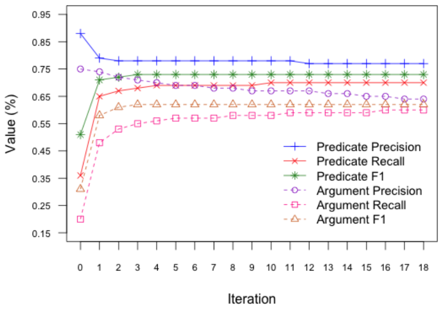


FIGURE 5.5: Values at each bootstrap iteration.

cutoff of 3. This combination of TF+RH filters, bootstrapping with $k_0 = 9$ and an iteration cutoff of 3 is used in the rest of the evaluation (Section 5.3), denoted as $\texttt{FB}_{best}$.

The initial evaluation of the proposed approach on $\texttt{French}_{\text{gold}}$ indicates that it successfully addresses issues of translation shift and outperforms earlier projection approaches in terms of precision and recall. In the next section, I use the best determined setup $\texttt{FB}_{best}$ to generate PROPBANKS for a comprehensive evaluation of the approach for a range of different languages.

5.3 Evaluation

In this section, I use the proposed method to generate PROPBANKS for 7 languages and comprehensively evaluate the generated resources. I seek to answer the following questions: (1) What is the estimated quality for the generated PROPBANKS? How well does the approach work without language-specific adaptation? (2) Are there notable differences in quality from language to language; if so, why? I also present initial investigations on how different factors affect the performance of the proposed method.

5.3.1 Experimental Setup

The experimental evaluation is challenging since there are no gold standard datasets against which the generated PROPBANKS can be automatically compared. The sole exception to this is $\texttt{French}_{\text{gold}}$, which was used in the initial experiments to determine parameterization. However, since I wish to evaluate in how far the approach generalizes, I require an evaluation methodology that can be applied to all languages. This section discusses the setup that was chosen.

LANGUAGE	DEPENDENCY PARSER	DATA SET	#SENTENCES
Arabic	STANFORD	UN	481K
Chinese	MATE-G	UN	2,986K
French	MATE-T	UN	2,542K
German	MATE-T	Europarl	560K
Hindi	MALT	Hindencorp	54K
Russian	MALT	UN	2,638K
Spanish	MATE-G	UN	2,304K

TABLE 5.4: Experimental setup.

Dependency parsers: STANFORD: [97], MATE-G: [99], MATE-T: [100], MALT: [98]. **Parallel corpora**: UN: [177], Europarl: [22], Hindencorp: [178]. **Word alignment**: The UN corpus is already word-aligned. For others, I use the Berkeley Aligner [179].

5.3.1.1 Resources for each TL

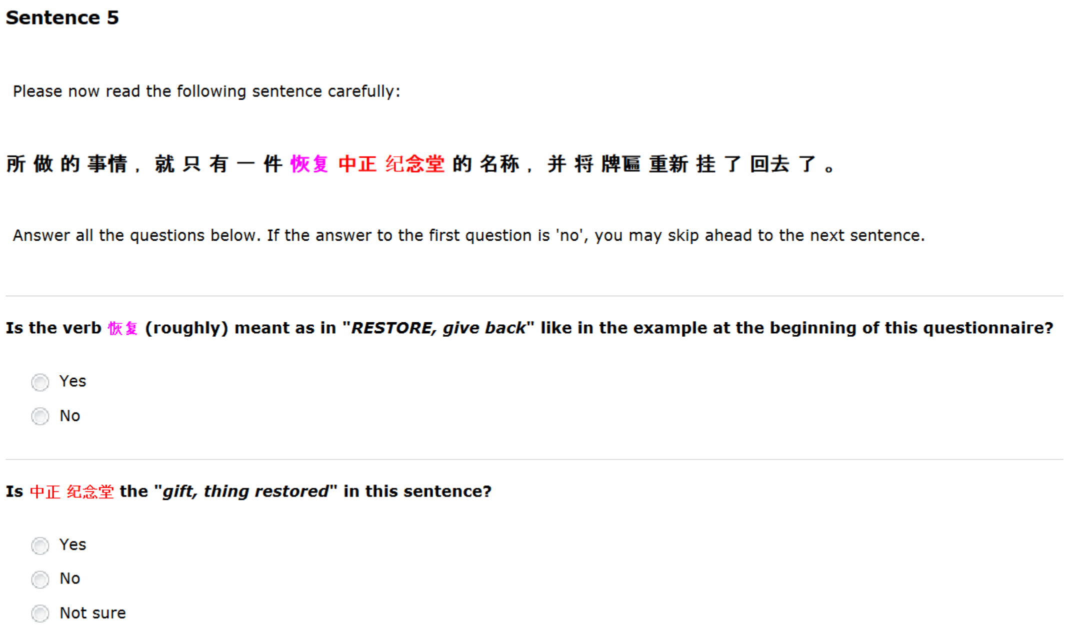
Table 5.4 lists the 7 different TLs evaluated in this section. I chose these TLs because (1) they are among top 10 most influential languages in the world [180]; and (2) I could find language experts to evaluate the results. English is used as SL in all the experiments. For each TL, I require a parallel corpus and a dependency parser.

The parallel corpus is used as dataset on which the projection approach is executed. For Arabic, Chinese, French, Russian and Spanish, I use the UN [177] corpus, which was automatically generated from United Nations documents. For German, I use Europarl, which was generated from documentation on parliamentary discussions in the European Union. Since for Hindi no such "governmental" parallel corpora exist, I use Hindencorp, a resource that was gathered from different resources such as the Web. From each parallel corpus, I only keep sentences that are considered well-formed based on a set of standard heuristics. For example, I require a well-formed sentence to end in punctuation and not to contain certain special characters. For Arabic, as the dependency parser I use has relatively poor parsing accuracy, I additionally require sentences to be shorter than 100 characters. Table 5.4 lists the total sizes of each parallel corpus and the subset of sentences used for evaluation.

A dependency parser is necessary for two reasons: On the one hand, the *Reattachment Heuristic* (see Section 5.2.1.2) requires dependency trees in order to ensure that argument labels are always the syntactic heads of their constituents. On the other hand, SRL has been shown to greatly benefit from deep syntactic analysis [] and SRL systems generally require dependency trees as input []. For each language, I used publicly available dependency parsers.

5.3.1.2 Evaluation Task

I execute the approach with setup $\texttt{FB}_{best}$ for each TL and extract all complete sentences to form the generated PROPBANKS. Since there is no gold annotated corpus available, I chose to conduct



Sentence 5

Please now read the following sentence carefully:

所 做 的 事情 ， 就 只 有 一 件 恢复 中正 纪念堂 的 名称 ， 并 将 牌匾 重新 挂 了 回去 了 。

Answer all the questions below. If the answer to the first question is 'no', you may skip ahead to the next sentence.

Is the verb 恢复 (roughly) meant as in "*RESTORE, give back*" like in the example at the beginning of this questionnaire?

○ Yes
○ No

Is 中正 纪念堂 the "*gift, thing restored*" in this sentence?

○ Yes
○ No
○ Not sure

FIGURE 5.6: Screenshot of the evaluation interface.

a manual evaluation for each TL, each executed identically: For each TL I randomly selected 100 complete sentences with their generated semantic labels and assigned them to two language experts who were instructed to evaluate the semantic labels (based on their English descriptions) for the predicates and their core arguments. For each label, they were asked to determine (1) whether the label is correct; (2) if yes, then whether the boundary of the labeled constituent is correct: If also yes, they were instructed to mark the label as *fully correct*, otherwise as *partially correct*.

The evaluation was conducted using the infrastructure of Amazon's Mechanical Turk. Sentences were grouped by frame and presented to users. For each frame group, users were first given an example sentence and its frame annotation in English. They then proceeded to evaluate sentences in the TL annotated with this frame. Figure 5.6 gives an example of this.

Metrics I used the standard measures of precision, recall, and F1 to measure the performance of the SRLs, with the following two schemes: (1) *Exact*: Only fully correct labels are considered as true positives; (2) *Partial*: Both fully and partially correct matches are considered as true positives.[7]

[7] Note that since the manually evaluated semantic labels are only a small fraction of the labels generated, the performance numbers obtained from manual evaluation is only an estimate of the actual quality for the generated resources.Thus the numbers obtained based on manual evaluation cannot be directly compared against the numbers computed over $\texttt{French}_{\text{gold}}$.

		PREDICATE			ARGUMENT				
LANGUAGE	Match	P	R	F1	P	R	F1	Agr	κ
Arabic	part.	0.97	0.89	0.93	0.86	0.69	0.77	0.92	0.87
	exact	0.97	0.89	0.93	0.67	0.63	0.65	0.85	0.77
Chinese	part.	0.97	0.88	0.92	0.93	0.83	0.88	0.95	0.91
	exact	0.97	0.88	0.92	0.83	0.81	0.82	0.92	0.86
French	part.	0.95	0.92	0.94	0.92	0.76	0.83	0.97	0.95
	exact	0.95	0.92	0.94	0.86	0.74	0.8	0.95	0.91
German	part.	0.96	0.92	0.94	0.95	0.73	0.83	0.95	0.91
	exact	0.96	0.92	0.94	0.91	0.73	0.81	0.92	0.86
Hindi	part.	0.91	0.68	0.78	0.93	0.66	0.77	0.94	0.88
	exact	0.91	0.68	0.78	0.58	0.54	0.56	0.81	0.69
Russian	part.	0.96	0.94	0.95	0.91	0.68	0.78	0.97	0.94
	exact	0.96	0.94	0.95	0.79	0.65	0.72	0.93	0.89
Spanish	part.	0.96	0.93	0.95	0.85	0.74	0.79	0.91	0.85
	exact	0.96	0.93	0.95	0.75	0.72	0.74	0.85	0.77

TABLE 5.5: Estimated precision and recall over seven languages.

5.3.2 Experiment 1: Quality of Generated PropBanks

Table 5.5 summarizes the estimated quality of semantic labels generated by the proposed method for all seven TLs. As can be seen, the proposed method performed well for all seven languages and generated high quality semantic labels across the board. For predicate labels, the precision is over 95% and the recall is over 85% for all languages except for Hindi. For argument labels, when considering partially correct matches, the precision is at least 85% (above 90% for most languages) and the recall is between 66% to 83% for all the languages. These encouraging results obtained from a diverse set of languages implies the generalizability of the proposed method. In addition, the inter-annotator agreement is very high for all the languages, indicating that the results obtained based on manual evaluation are very reliable.

In addition, I make a number of interesting observations:

Dependency Parsing Accuracy The precision for exact argument labels is significantly below partial matches, particularly for Hindi (↓35 pp) and Arabic (↓19 pp). Since argument boundaries are determined syntactically, such errors are caused by dependency parsing. The fact that Hindi and Arabic suffer the most from this issue is consistent with the poorer performance of their dependency parsers compared to other languages [97, 98].

Hindi as the Main Outlier The results for Hindi are much worse than the results for other languages. Besides the poorer dependency parser performance, the size of the parallel corpus used could be a factor: `Hindencorp` is one to two orders of magnitude smaller

	Predicate			Argument		
Sample Size	P	R	F1	P	R	F1
100%	0.87	0.81	0.84	0.86	0.74	0.8
10%	0.88	0.8	0.84	0.87	0.72	0.79
1%	0.9	**0.76**	0.83	0.89	**0.67**	0.76

Table 5.6: Estimated impact of downsampling parallel corpus.

	Predicate			Argument		
Heuristic	P	R	F1	P	R	F1
none*	0.87	0.81	0.84	0.86	0.74	0.8
none**	0.88	0.8	0.84	**0.76**	**0.65**	0.7
customization*	0.87	0.81	0.84	**0.9**	0.74	0.81

Table 5.7: Impact of English SRLs (*=ClearNLP, **=MATE-SRL) and language-spec. customization (*filter synt. expletive*).

than the other corpora. The quality of the parallel corpus could be a reason as well: `Hindencorp` was collected from various sources, while both `UN` and `Europarl` were extracted from governmental proceedings.

Language-specific Errors Certain errors occur more frequently in some languages than others. An example are deverbal nouns in Chinese [110] in formal passive constructions with support verb Since I currently only consider verbs for predicate labels, predicate labels are projected onto the support verbs instead of the deverbal nouns. Such errors appear for light verb constructions in all languages, but particularly affect Chinese due to the high frequency of this passive construction in the `UN` corpus.

Low Fraction of Complete Sentences As Table 5.8 shows, the fraction of complete sentences in the generated PropBanks is rather low, indicating the impact of moderate recall on the size of generated PropBanks. Especially for languages for which only small parallel corpora are available, such as Hindi, this points to the need to address recall issues in future work.

I proceed to evaluate in more detail the impact of additional factors on the quality of generated resources.

5.3.3 Experiment 2: Impact of Additional Factors

The observations made in Section 5.3.2 suggests a few factors that may potentially affect the performance of the proposed method. To better understand their impact, I conducted the following initial investigation. SRL models produced in this set of experiments were evaluated

PROPBANK	#COMPLETE	%COMPLETE	#VERBS
Arabic	68.512	14%	330
Chinese	419,140	14%	1,102
French	248.256	10%	1145
German	44.007	8%	537
Hindi	1.623	3%	59
Russian	496.033	19%	1.349
Spanish	165.582	7%	909

TABLE 5.8: Characteristics of the generated PROPBANKS.

using $\texttt{French}_{\text{gold}}$, sampled and evaluated in the same way as other experiments in this section for comparability. I looked at the following factors:

Data Size I varied the data size for French by downsampling the UN corpus. As one can see from Table 5.6, downsampling the dataset by one order of magnitude (to 250k sentences) only slightly affects precision, while downsampling to 25k sentences has a more pronounced but still small impact on recall. It appears that data size does not have significant impact on the performance of the proposed method.

Language-specific Customizations While the proposed method is language-agnostic, intuitively language-specific customization can be helpful in addressing language-specific errors. As an initial experiment, I address one type of common errors for French that involves the syntactic expletive "*il*" [181] in "existential there" constructions such as "*il faut*". As Figure 5.2 showed, the expletive "*il*" is wrongly labeled with with role information. I manually define a heuristic to filter out such projections. As shown in Table 5.7, this simple customization results in a small increase in precision, suggesting that language-specific customization can be helpful.

Quality of English SRL As noted in Section 5.2.1.3, errors made by English SRL are often prorogated to the TL via projection. To assess the impact of English SRL quality, I used two different English SRL systems: CLEARNLP and MATE-SRL. As can be seen from Table 5.7, the impact of English SRL quality is substantial on argument labeling.

These experiments indicate that especially the latter two factors, i.e. language-specific customizations and English SRL quality, hold potential for further increasing the quality of generated PROPBANKS with the proposed approach. In the next section, I discuss these results and identify possible avenues for future research.

5.4 Discussion and Summary

The evaluation shows that the proposed two-staged method to construct multilingual SRL resources using monolingual SRL and parallel data works well across different languages without any language specific customization. To facilitate future research on multilingual SRL, I release the created PROPBANKS for all 7 languages to the research community to encourage further research. Table 5.8 gives an overview over the resources.

A number of issues remain that future work may address. First, one noted problem is that the subset of complete sentences in the generated resources is small compared to the input parallel corpora. As Table 5.8 illustrates, the subset of complete sentences ranges from 3% for Hindi to 19% for Russian. Since the goal is to have a broad coverage of semantic frames, future work may investigate methods to improve recall. In addition, the precision of the generated resources, ranging from 85% to 95% in non-strict setting (see Table 5.5) can be further improved by addressing some of the factors that were noted during the experimental evaluation. In more detail, I identify the following principal avenues for further research:

Languge-specific Studies An initial experiment has shown that language-specific customizations can increase the quality of generated resources. One avenue for future research could be to investigate this further by focusing on a small set of languages and determining language-specific sources of errors and heuristics to counteract such error sources. One interesting result of such research could be to gain insight on an “upper bound” of the quality of resource that annotation projection with highly specific customizations can produce.

Semantic Role Labeling Initial experiments have also shown a strong impact of source language SRL quality: Not only are errors in source language SRL the most common unhandled error class (see Section 5.2.1.3), but also have experiments shown that switching SRL systems resulted in an absolute decrease of 10 f-measure percentage points (Table5.7). This means that increasing the quality of English SRL will greatly benefit a projection approach. Consequently, an avenue for future research may be to work more specifically on creating better SRL systems, or combining the output of several SRL systems to produce better English SRL [182].

Dependency Parsing Another crucial bottleneck is the need for good dependency parsing for each language. While dependency parsers were available for all 7 languages that I investigated, there are many underresourced languages for which such resources are not available [183]. Consequently, one avenue for future research may investigate in how far the proposed approach works for languages in which no dependency parsers are available. A possible outcome of such research could be strategies that make the need for dependency parsing obsolete.

Other Types of SRL While the experiments so far have focused on frames evoked by verbs, projects such as NOMBANK [184] also focus on frames evoked by nouns. An interesting avenue for future work would be to include such frames in the projection approach.

I conclude with the observation that the proposed method outperforms previous approaches in both precision and recall, and that for the first time annotation projection was used to generate resources for a range of different languages from three language families. Since all languages use the same set of frame labels as the English PROPBANK, this enables the training of semantic parsers that parse different languages into a broad-coverage language-neutral representation of shallow semantics. As I argue in the next and final chapter, this representation may be expanded to additional IE tasks in future work.

6

Conclusion

Answers are like opium: the more you imbibe, the more you need. Which is why the sober man finds solace in mystery.

– Excerpt from [2]

6.1 Summary

Practical scenarios for Relation Extraction are often characterized by the availability of large, multilingual data of unknown content on the one hand and vague and shifting information needs on the other. This thesis researched methods for Relation Extraction in such scenarios: Based on an investigation of unsupervised Relation Discovery, I proposed the novel paradigm of *Exploratory Relation Extraction*. It formulates a user-driven but data-guided process of *exploration* for relations of interest in unknown data. I showed how distributional evidence and an informed linguistic abstraction can be employed to allow users to openly explore a dataset for relations of interest and rapidly prototype extractors for discovered relations at minimal effort.

For multilingual data, I proposed the use of a *language-neutral representation of shallow semantics*. This representation enables a shared feature space for different languages against which extractors can be developed. I presented a novel approach which expands English-language Semantic Role Labeling (SRL) to other languages. I used this approach to generate multilingual SRL resources for 7 distinct languages from different language groups, namely Arabic, Chinese, French, German, Hindi, Russian and Spanish in order to bootstrap high quality semantic parsers for these languages.

Together, the researched approaches represent a novel way for data scientists to work with large multilingual datasets of unknown content.

6.2 Outlook

The future of text mining is bright. While this thesis focused specifically on the task of extracting binary relations from text, future work could apply these methods to further Information Extraction tasks and domains. An example of this are *entity-level* extraction tasks: It would be interesting to expand ERE to enable users to discover new entity types in a dataset, allowing even greater freedom of exploration on part of the data scientist. This would make ERE applicable to a multitude of practical applications in specific domains such as biomedical or legal text data and enable prototyping and construction of topic-specific knowledge bases.

The road towards an interlingua. The entity-level view is also especially interesting with regards to the proposed language-neutral representation, since entities can be expressed differently from language to language. For instance, the city of "*Milan*" is called "*Milano*" in Italian and "*Mailand*" in German. This indicates that in order to expand this representation to the entity-level, one requires a step of linking multilingual entity mentions to a knowledge base. Similarly, other IE tasks may require additional extensions to the representation. I believe that future work should extend the proposed language-neutral representation in such an incremental, task-by-task fashion. Such a "bottom-up" approach is in my opinion a realistic method of arriving at a practical and multi-purpose interlingual representation of shallow semantics. I believe that advances in SRL, knowledge base construction and large-scale data mining have placed the elusive idea of an interlingua within our grasp like never before.

Bibliography

[1] S. Bakker. *The Warriot Prophet*. Overlook Press, 2005.

[2] S. Bakker. *The Darkness That Comes Before*. Overlook Press, 2004.

[3] Josh James. Data never sleeps 2.0. https://www.domo.com/blog/2014/04/data-never-sleeps-2-0/, 2014. Accessed: 2015-05-15.

[4] Michelle Weber. Im(press)ive! your year in review. https://en.blog.wordpress.com/2015/01/06/2014-in-review/, 2015. Accessed: 2015-05-15.

[5] Mediawiki Foundation. Wikipedia:statistics. http://en.wikipedia.org/wiki/Wikimedia_project, 2015. Accessed: 2015-05-15.

[6] John Gantz and David Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the Future*, 2007: 1–16, 2012.

[7] Ellen Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049, 1996.

[8] Jonas Rest. In berlin wächst deutschlands neuer wirtschaftsmotor heran. http://www.berliner-zeitung.de/wirtschaft/boomende-start-up-branche-in-berlin-waechst-deutschlands-neuer-wirtschaftsmotor-heran,10808230,28382726.html, 2013. Accessed: 2015-05-15.

[9] Laura Chiticariu, Yunyao Li, and Frederick R Reiss. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 827–832, 2013.

[10] Ce Zhang, Feng Niu, Christopher Ré, and Jude Shavlik. Big data versus the crowd: Looking for relationships in all the right places. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 825–834. Association for Computational Linguistics, 2012.

[11] Hans Uszkoreit. Learning relation extraction grammars with minimal human intervention: strategy, results, insights and plans. In *Computational Linguistics and Intelligent Text Processing*, pages 106–126. Springer, 2011.

[12] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL/AFNLP*, pages 1003–1011, 2009.

[13] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Relation adaptation: Learning to extract novel relations with minimum supervision. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[14] Frederick Reiss, Sriram Raghavan, Rajasekar Krishnamurthy, Huaiyu Zhu, and Shivakumar Vaithyanathan. An algebraic approach to rule-based information extraction. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 933–942. IEEE, 2008.

[15] Yunyao Li, Laura Chiticariu, Huahai Yang, Frederick R Reiss, and Arnaldo Carreno-fuentes. Wizie: a best practices guided development environment for information extraction. In *Proceedings of the ACL 2012 System Demonstrations*, pages 109–114. Association for Computational Linguistics, 2012.

[16] Sonal Gupta and Christopher D Manning. Spied: Stanford pattern-based information extraction and diagnostics. *Sponsor: Idibon*, page 38, 2014.

[17] Ralph Grishman and Yifan He. An information extraction customizer. In *Text, Speech and Dialogue*, pages 3–10. Springer, 2014.

[18] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423. Association for Computational Linguistics, 2004.

[19] Jannik Strötgen and Michael Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324. Association for Computational Linguistics, 2010.

[20] Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. Cross-lingual information extraction system evaluation. In *Proceedings of the 20th international Conference on Computational Linguistics*, page 882. Association for Computational Linguistics, 2004.

[21] The Lemur Project. The clueweb09 dataset. `http://www.lemurproject.org/clueweb09.php/`, 2015. Accessed: 2015-05-15.

[22] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

[23] Mediawiki Foundation. List of wikipedias. http://en.wikipedia.org/wiki/List_of_Wikipedias, 2015. Accessed: 2015-05-15.

[24] Daniel Pimienta, Daniel Prado, and Álvaro Blanco. *Twelve years of measuring linguistic diversity in the Internet: balance and perspectives*. United Nations Educational, Scientific and Cultural Organization, 2009.

[25] B. Rosenfeld and R. Feldman. Clustering for unsupervised relation identification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 411–418. ACM, 2007.

[26] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Relational duality: unsupervised extraction of semantic relations between entities on the web. In *WWW*, pages 151–160, 2010.

[27] Yusuke Shinyama and Satoshi Sekine. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311. Association for Computational Linguistics, 2006.

[28] Anders Björkelund, Love Hafdell, and Pierre Nugues. Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 43–48. Association for Computational Linguistics, 2009.

[29] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106, 2005.

[30] Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics, 2009.

[31] K. Erk, A. Kowalski, S. Pado, and S. Pinkal. Towards a resource for lexical semantics: A large german corpus with extensive semantic annotation. In *ACL*, 2003.

[32] Wajdi Zaghouani, Mona Diab, Aous Mansouri, Sameer Pradhan, and Martha Palmer. The revised arabic propbank. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 222–226. Association for Computational Linguistics, 2010.

[33] Ashwini Vaidya, Jinho D Choi, Martha Palmer, and Bhuvana Narasimhan. Analysis of the hindi proposition bank using dependency structure. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 21–29. Association for Computational Linguistics, 2011.

[34] A. Akbik, L. Visengeriyeva, P. Herger, H. Hemsen, and A. Löser. Unsupervised discovery of relations and discriminative extraction patterns. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 17–32, 2012.

[35] A. Akbik, L. Visengeriyeva, J. Kirschnick, and A. Löser. Effective selectional restrictions for unsupervised relation extraction. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, 2013.

[36] Alan Akbik and Larysa Visengeriyeva. Proceedings of the first aha!-workshop on information discovery in text. In *24th International Conference on Computational Linguistics*, 2014.

[37] Alan Akbik, Thilo Michael, and Christoph Boden. Exploratory relation extraction in large text corpora. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 2087–2096, 2014. URL http://aclweb.org/anthology/C/C14/C14-1197.pdf.

[38] Thilo Michael and Alan Akbik. SCHNÄPPER: A web toolkit for exploratory relation extraction. In *ACL 2015, 53rd Annual Meeting of the Association for Computational Linguistics Beijing, China*, page to appear, 2015.

[39] Alan Akbik, Oresti Konomi, and Michail Melnikov. Propminer: A workflow for interactive information extraction and exploration using dependency trees. In *ACL System Demonstrations*. Association for Computational Linguistics, 2013.

[40] Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. Generating high quality proposition banks for multilingual semantic role labeling. In *ACL 2015, 53rd Annual Meeting of the Association for Computational Linguistics Beijing, China*, page to appear, 2015.

[41] Ralph Grishman. Information extraction. *The Handbook of Computational Linguistics and Natural Language Processing*, pages 515–530, 2003.

[42] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM, 2000.

[43] Sergey Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pages 172–183. Springer, 1999.

[44] Ellen Riloff et al. Automatically constructing a dictionary for information extraction tasks. In *AAAI*, pages 811–816, 1993.

[45] A. Akbik and A. Löser. Kraken: N-ary facts in open information extraction. In *AKBC-WEKEX*, pages 52–56. Association for Computational Linguistics, 2012.

[46] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366. International World Wide Web Conferences Steering Committee, 2013.

[47] Xiao Ling and Daniel S. Weld. Temporal information extraction. In *24th. AAAI Conference on Artificial Intelligence*, 2010.

[48] Beth M Sundheim. Plans for a task-oriented evaluation of natural language understanding systems. In *Proceedings of the workshop on Speech and Natural Language*, pages 197–202. Association for Computational Linguistics, 1989.

[49] Beth M Sundheim. Overview of the third message understanding evaluation and conference. In *Proceedings of the 3rd conference on Message understanding*, pages 3–16. Association for Computational Linguistics, 1991.

[50] Douglas E Appelt. Introduction to information extraction. *Ai Communications*, 12(3): 161–172, 1999.

[51] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. doi: 10.3115/992628.992709. URL http://dx.doi.org/10.3115/992628.992709.

[52] Lynette Hirschman. Comparing muck-ii and muc-3: Assessing the difficulty of different tasks. In *Proceedings of the 3rd conference on Message understanding*, pages 25–30. Association for Computational Linguistics, 1991.

[53] Ralph Grishman and John Sterling. Preference semantics for message understanding. In *Proceedings of the workshop on Speech and Natural Language*, pages 71–74. Association for Computational Linguistics, 1989.

[54] Beth M Sundheim. Overview of the fourth message understanding evaluation and conference. In *Proceedings of the 4th conference on Message understanding*, pages 3–21. Association for Computational Linguistics, 1992.

[55] Beth M Sundheim. Tipster/muc-5: information extraction system evaluation. In *Proceedings of the 5th conference on Message understanding*, pages 27–44. Association for Computational Linguistics, 1993.

[56] Libin Shen and Jinying Chen. Using supertag in muc-7 template relation task. Technical report, Citeseer.

[57] Ralph Grishman and Beth Sundheim. *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*. Morgan Kaufmann Publishers, Inc, 1995.

[58] Ralph Grishman and Beth Sundheim. Design of the muc-6 evaluation. In *Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996*, pages 413–422. Association for Computational Linguistics, 1996.

[59] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation.

[60] Stephanie Strassel, Mark A Przybocki, Kay Peterson, Zhiyi Song, and Kazuaki Maeda. Linguistic resources and evaluation techniques for evaluation of cross-document automatic content extraction. In *LREC*, 2008.

[61] Ralph Grishman. Message understanding conference proceedings muc-7. URL http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/overview.html,accessed05/05/2015.

[62] Razvan C Bunescu and Raymond J Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics, 2005.

[63] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics, 2005.

[64] Jing Jiang and ChengXiang Zhai. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120, 2007.

[65] Raymond J Mooney and Razvan C Bunescu. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178, 2005.

[66] Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *EACL*, volume 18, pages 401–408. Citeseer, 2006.

[67] Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick R Reiss, and Shivakumar Vaithyanathan. Systemt: an algebraic approach to declarative information extraction. In *ACL*, pages 128–137. Association for Computational Linguistics, 2010.

[68] Witold Drozdzynski, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu. Shallow processing with unification and typed feature structures — foundations and applications. *Künstliche Intelligenz*, 1:17–23, 2004. URL http://www.kuenstliche-intelligenz.de/archiv/2004_1/sprout-web.pdf.

[69] Peter Kluegl, Martin Atzmueller, and Frank Puppe. Textmarker: A tool for rule-based information extraction. In *Proceedings of the Biennial GSCL Conference*, pages 233–240, 2009.

[70] Dayne Freitag. Information extraction from html: Application of a general machine learning approach. In *AAAI/IAAI*, pages 517–523, 1998.

[71] Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine learning*, 34(1-3):233–272, 1999.

[72] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, and Michael Young. Machine Learning: The High Interest Credit Card of Technical Debt. In *SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop)*, 2014.

[73] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

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

[75] Mark A Greenwood and Mark Stevenson. Improving semi-supervised acquisition of relation extraction patterns. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 29–35. Association for Computational Linguistics, 2006.

[76] Fei-Yu Xu. *Bootstrapping Relation Extraction from Semantic Seeds*. PhD thesis, Saarland University, 2008.

[77] Feiyu Xu, Hans Uszkoreit, Sebastian Krause, and Hong Li. Boosting relation extraction with limited closed-world knowledge. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1354–1362. Association for Computational Linguistics, 2010.

[78] A. Sun and R. Grishman. Semi-supervised semantic pattern discovery with guidance from unsupervised pattern clusters. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1194–1202. Association for Computational Linguistics, 2010.

[79] Fei Wu and Daniel S Weld. Autonomously semantifying wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 41–50. ACM, 2007.

[80] Mihai Surdeanu, David McClosky, Julie Tibshirani, John Bauer, Angel X Chang, Valentin I Spitkovsky, and Christopher D Manning. A simple distant supervision approach for the tac-kbp slot filling task. In *Proceedings of Text Analysis Conference 2010 Workshop*. Citeseer, 2010.

[81] Sebastian Krause, Hong Li, Hans Uszkoreit, and Feiyu Xu. Large-scale learning of relation-extraction rules with distant supervision from the web. In *The Semantic Web–ISWC 2012*, pages 263–278. Springer, 2012.

[82] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *HLT-NAACL*, pages 777–782, 2013.

[83] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. Yago2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 229–232, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0637-9. doi: 10.1145/1963192.1963296. URL http://doi.acm.org/10.1145/1963192.1963296.

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

[85] George Krupka, Paul Jacobs, Lisa Rau, and Lucja Iwańska. Ge: Description of the nltoolset system as used for muc-3. In *Proceedings of the 3rd conference on Message understanding*, pages 144–149. Association for Computational Linguistics, 1991.

[86] Ralph Grishman and John Sterling. New york university: Description of the proteus system as used for muc-5. In *Proceedings of the 5th conference on Message understanding*, pages 181–194. Association for Computational Linguistics, 1993.

[87] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011.

[88] Wendy Lehnert, Claire Cardie, David Fisher, Ellen Riloff, and Robert Williams. University of massachusetts: Description of the circus system as used for muc-3. In *Proceedings*

of the 3rd conference on Message understanding, pages 223–233. Association for Computational Linguistics, 1991.

[89] Damaris Ayuso, Sean Boisen, Heidi Fox, Herb Gish, Robert Ingria, and Ralph Weischedel. Bbn: Description of the plum system as used for muc-4. In *Proceedings of the 4th conference on Message understanding*, pages 169–176. Association for Computational Linguistics, 1992.

[90] Peter D. Turney. Expressing implicit semantic relations without supervision. In *ACL*, 2006.

[91] Wei Wang, Romaric Besançon, Olivier Ferret, and Brigitte Grau. Filtering and clustering relations for unsupervised information extraction in open domain. In *CIKM*, pages 1405–1414, 2011.

[92] A. Akbik and J. Bross. Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *1st. Workshop on Semantic Search at 18th. WWW Conference*, 2009.

[93] L Tesnière. *Elements de syntaxe structurale*. Editions Klincksieck, 1959.

[94] Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The conll 2007 shared task on dependency parsing. In *Proceedings of the CoNLL shared task session of EMNLP-CoNLL*, pages 915–932. sn, 2007.

[95] Sabine Buchholz and Erwin Marsi. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164. Association for Computational Linguistics, 2006.

[96] Jinho D. Choi and Andrew McCallum. Transition-based dependency parsing with selectional branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.

[97] Spence Green and Christopher D Manning. Better arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 394–402. Association for Computational Linguistics, 2010.

[98] Joakim Nivre, Johan Hall, and Jens Nilsson. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, pages 2216–2219, 2006.

[99] Bernd Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *COLING*, pages 89–97. Association for Computational Linguistics, 2010.

[100] Bernd Bohnet and Joakim Nivre. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint*

Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1455–1465. Association for Computational Linguistics, 2012.

[101] Xavier Carreras and Lluís Màrquez. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 152–164. Association for Computational Linguistics, 2005.

[102] Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. An analysis of open information extraction based on semantic role labeling. In *K-CAP*, pages 113–120, 2011.

[103] Beth Levin. *English verb classes and alternations: A preliminary investigation*. University of Chicago press, 1993.

[104] Dan Shen and Mirella Lapata. Using semantic roles to improve question answering. In *EMNLP-CoNLL*, pages 12–21. Citeseer, 2007.

[105] Umar Maqsud, Sebastian Arnold, Michael Hülfenhaus, and Alan Akbik. Nerdle: Topic-specific question answering using wikia seeds. In Lamia Tounsi and Rafal Rak, editors, *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference System Demonstrations, August 23-29, 2014, Dublin, Ireland*, pages 81–85. ACL, 2014. ISBN 978-1-941643-27-3. URL http://aclweb.org/anthology/C/C14/C14-2018.pdf.

[106] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

[107] Ana-Maria Giuglea and Alessandro Moschitti. Semantic role labeling via framenet, verbnet and propbank. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 929–936. Association for Computational Linguistics, 2006.

[108] Claire Bonial, Kevin Stowe, and Martha Palmer. Renewing and revising semlink. In *The GenLex Workshop on Linked Data in Linguistics*, 2013.

[109] Lonneke Van der Plas, Paola Merlo, and James Henderson. Scaling up automatic cross-lingual semantic role annotation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 299–304. Association for Computational Linguistics, 2011.

[110] Nianwen Xue. Semantic role labeling of nominalized predicates in chinese. In *Proceedings of the main conference on Human Language Technology Conference of the North*

American Chapter of the Association of Computational Linguistics, pages 431–438. Association for Computational Linguistics, 2006.

[111] P.D. Turney. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33(1):615–655, 2008.

[112] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*, 2012.

[113] Limin Yao, Sebastian Riedel, and Andrew McCallum. Unsupervised relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 712–720. Association for Computational Linguistics, 2012.

[114] Alexander Rush and Slav Petrov. Vine pruning for efficient multi-pass dependency parsing. In *NAACL '12*, 2012.

[115] Yue Zhang and Joakim Nivre. Transition-based dependency parsing with rich non-local features. In *ACL (Short Papers)*, pages 188–193, 2011.

[116] Zellig S Harris. *Distributional structure*. Springer, 1970.

[117] L. Wittgenstein. *Philosophische Untersuchungen*. Blackwell, 1953.

[118] P.D. Turney. Analogy perception applied to seven tests of word comprehension. *Journal of Experimental & Theoretical Artificial Intelligence*, 23(3):343–362, 2011.

[119] Dekang Lin and Patrick Pantel. Dirt: discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM, 2001.

[120] Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of english words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 183–190. Association for Computational Linguistics, 1993.

[121] JR Firth. A synopsis of linguistic theory 1930-1955, volume 1952-59. the philological society, 1957.

[122] D. Lin and X. Wu. Phrase clustering for discriminative learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1030–1038, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-46-6. URL http://dl.acm.org/citation.cfm?id=1690219.1690290.

[123] D. Lin, K. Church, H. Ji, S. Sekine, D. Yarowsky, S. Bergsma, K. Patil, E. Pitler, R. Lathbury, V. Rao, et al. New tools for web-scale n-grams. In *Proceedings of LREC*, 2010.

[124] G. Zhou, J. Zhao, K. Liu, and L. Cai. Exploiting web-derived selectional preference to improve statistical dependency parsing. In *Proceedings of ACL*, pages 1556–1565, 2011.

[125] O. Täckström, R. McDonald, and J. Uszkoreit. Cross-lingual word clusters for direct transfer of linguistic structure. *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2012.

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

[127] M.C. De Marneffe, B. MacCartney, and C.D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[128] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. Open information extraction: The second generation. In *IJCAI*, pages 3–10, 2011.

[129] F. Mesquita, Y. Merhav, and D. Barbosa. Extracting information networks from the blogosphere: State-of-the-art and challenges. In *Fourth Int'l AAAI conference on weblogs and social media*, 2010.

[130] Tuan Anh Do. Gezieltes retrieval von faktenstarken sätzen im web auf basis von wikipedia. B.S. Thesis, Technische Universität Berlin, 2013.

[131] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[132] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[133] Yoshiki Niwa and Yoshihiko Nitta. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 304–309. Association for Computational Linguistics, 1994.

[134] J.A. Bullinaria and J.P. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526, 2007.

[135] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[136] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[137] Davoud Moulavi, Pablo A Jaskowiak, RJGB Campello, Arthur Zimek, and Jörg Sander. Density-based clustering validation. In *Proceedings of the 14th SIAM International Conference on Data Mining (SDM), Philadelphia, PA*, 2014.

[138] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486, August 2009. ISSN 1386-4564. doi: 10.1007/s10791-008-9066-8. URL http://dx.doi.org/10.1007/s10791-008-9066-8.

[139] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[140] S. Bakker. *The White-Luck Warrior*. Overlook Press, 2011.

[141] AnHai Doan, Jeffrey F Naughton, Raghu Ramakrishnan, Akanksha Baid, Xiaoyong Chai, Fei Chen, Ting Chen, Eric Chu, Pedro DeRose, Byron Gao, et al. Information extraction challenges in managing unstructured data. *ACM SIGMOD Record*, 37(4):14–20, 2009.

[142] Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. Open language learning for information extraction. In *EMNLP-CoNLL*, pages 523–534, 2012.

[143] Marjorie Freedman, Lance Ramshaw, Elizabeth Boschee, Ryan Gabbard, Gary Kratkiewicz, Nicolas Ward, and Ralph Weischedel. Extreme extraction: machine reading in a week. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1437–1446. Association for Computational Linguistics, 2011.

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

[145] Ryen W White and Resa A Roth. Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1): 1–98, 2009.

[146] Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen 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*, pages 25–26. Association for Computational Linguistics, 2007.

[147] Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics, 2012.

[148] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume One*, pages 3–10. AAAI Press, 2011.

[149] Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450. ACM, 2010.

[150] Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. FACC1: freebase annotation of ClueWeb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0), 2013.

[151] Jinho D Choi and Andrew McCallum. Transitionbased dependency parsing with selectional branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria*, 2013.

[152] Dave Orr. 50,000 lessons on how to read: a relation extraction corpus. http://googleresearch.blogspot.de/2013/04/50000-lessons-on-how-to-read-relation.html, 2013. Accessed: 2015-05-15.

[153] Chang Wang, Aditya Kalyanpur, James Fan, Branimir K Boguraev, and DC Gondek. Relation extraction and scoring in deepqa. *IBM Journal of Research and Development*, 56(3.4):9–1, 2012.

[154] S. Bakker. *The Judging Eye*. Overlook Press, 2009.

[155] Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. An annotation scheme for free word order languages. In *Proceedings of the fifth conference on Applied natural language processing*, pages 88–95. Association for Computational Linguistics, 1997.

[156] Martha Palmer, Nianwen Xue, Olga Babko-Malaya, Jinying Chen, and Benjamin Snyder. A parallel proposition bank ii for chinese and english. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 61–67. Association for Computational Linguistics, 2005.

[157] Vangelis Karkaletsis, Constantine D Spyropoulos, Claire Grover, M Pazienza, Jose Coch, and Dimitris Souflis. A platform for cross-lingual, domain and user adaptive web information extraction. In *ECAI*, volume 16, page 725, 2004.

[158] Kalina Bontcheva, Diana Maynard, Valentin Tablan, and Hamish Cunningham. Gate: A unicode-based infrastructure supporting multilingual information extraction. *Proceedings on Information Extraction for Slavonic and other Central and Eastern European Languages, Borovets, Bulgaria*, 2003.

[159] Hiroshi Uchida. 4• 2 atlas ii: A machine translation system using conceptual structure as an interlingua. In *Machine Translation Summit*, page 93. IOS Press, 1989.

[160] Teruko Mitamura, Eric H Nyberg, and Jaime G Carbonell. An efficient interlingua translation system for multi-lingual document production. In *Proceedings of Machine Translation Summit III*, 1991.

[161] Eduard Hovy and Sergei Nirenburg. Approximating an interlingua in a principled way. In *Proceedings of the workshop on Speech and Natural Language*, pages 261–266. Association for Computational Linguistics, 1992.

[162] Lori S Levin, Donna Gates, Alon Lavie, and Alex Waibel. An interlingua based on domain actions for machine translation of task-oriented dialogues. In *ICSLP*, volume 98, pages 1155–1158, 1998.

[163] Sebastian Pado. *Cross-Lingual Annotation Projection Models for Role-Semantic Information*. PhD thesis, Saarland University, 2007. MP.

[164] Paola Monachesi, Gerwert Stevens, and Jantine Trapman. Adding semantic role annotation to a corpus of written dutch. In *Proceedings of the Linguistic Annotation Workshop*, LAW '07, pages 77–84, 2007.

[165] Sebastian Padó and Mirella Lapata. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36(1):307–340, 2009.

[166] Roberto Basili, Diego De Cao, Danilo Croce, Bonaventura Coppola, and Alessandro Moschitti. Cross-language frame semantics transfer in bilingual corpora. In *Computational Linguistics and Intelligent Text Processing*, pages 332–345. Springer, 2009.

[167] Paola Merlo and Lonneke van der Plas. Abstraction and generalisation in semantic role labels: Propbank, verbnet or both? In *ACL 2009*, pages 288–296, 2009.

[168] Lonneke Van der Plas, Marianna Apidianaki, Rue John von Neumann, and Chenhua Chen. Global methods for cross-lingual semantic role and predicate labelling. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1279–1290. Association for Computational Linguistics, 2014.

[169] David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pages 1–8. Association for Computational Linguistics, 2001.

[170] Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(03):311–325, 2005.

[171] Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. Dependency grammar induction via bitext projection constraints. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 369–377. Association for Computational Linguistics, 2009.

[172] Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12, 2013.

[173] Ellen Riloff, Charles Schafer, and David Yarowsky. Inducing information extraction systems for new languages via cross-language projection. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[174] Seokhwan Kim, Minwoo Jeong, Jonghoon Lee, and Gary Geunbae Lee. A cross-lingual annotation projection approach for relation detection. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 564–571. Association for Computational Linguistics, 2010.

[175] Lonneke Van der Plas, Tanja Samardžić, and Paola Merlo. Cross-lingual validity of propbank in the manual annotation of french. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 113–117. Association for Computational Linguistics, 2010.

[176] Karin Kipper Schuler. *Verbnet: A Broad-coverage, Comprehensive Verb Lexicon*. PhD thesis, University of Pennsylvania, 2005.

[177] Alexandre Rafalovitch, Robert Dale, et al. United nations general assembly resolutions: A six-language parallel corpus. In *Proceedings of the MT Summit*, volume 12, pages 292–299, 2009.

[178] Ondřej Bojar, Vojtěch Diatka, Pavel Rychlỳ, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, Daniel Zeman, et al. Hindencorp–hindi-english and hindi-only corpus for machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, 2014.

[179] John DeNero and Percy Liang. The Berkeley Aligner. http://code.google.com/p/berkeleyaligner/, 2007.

[180] George Weber. Top languages: The world's 10 most influential languages. *Language Today*, December 1997.

[181] Laurence Danlos. Automatic recognition of french expletive pronoun occurrences. In *Natural language processing. Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 73–78. Citeseer, 2005.

[182] Daniel Zeman and Zdeněk Žabokrtskỳ. Improving parsing accuracy by combining diverse dependency parsers. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 171–178. Association for Computational Linguistics, 2005.

[183] Dan Klein and Christopher D Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 478. Association for Computational Linguistics, 2004.

[184] Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. Annotating noun argument structure for nombank. In *LREC*, volume 4, pages 803–806, 2004.