Bootstrapping Relation Extraction from Semantic Seeds

Fei-Yu Xu

A DISSERTATION
SUBMITTED TO THE PHILOSOPHY FACULTY
OF SAARLAND UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

SUPERVISED BY:
PROF. DR. HANS USZKOREIT

Abstract

Information Extraction (IE) is a technology for localizing and classifying pieces of relevant information in unstructured natural language texts and detecting relevant relations among them. This thesis deals with one of the central tasks of IE, i.e., relation extraction. The goal is to provide a general framework that automatically learns mappings between linguistic analyses and target semantic relations, with minimal human intervention. Furthermore, this framework is supposed to support the adaptation to new application domains and new relations with various complexities.

The central result is a new approach to relation extraction which is based on a minimally supervised method for automatically learning extraction grammars from a large collection of parsed texts, initialized by some instances of the target relation, called *semantic seed*. Due to the semantic seed approach, the framework can accommodate new relation types and domains with minimal effort. It supports relations of different arity as well as their projections. Furthermore, this framework is general enough to employ any linguistic analysis tools that provide the required type and depth of analysis.

The adaptability and the scalability of the framework is facilitated by the ***DARE*** rule representation model which is recursive and compositional. In comparison to other IE rule representation models, e.g., Stevenson and Greenwood (2006), the ***DARE*** rule representation model is expressive enough to achieve good coverage of linguistic constructions for finding mentions of the target relation. The powerful ***DARE*** rules are constructed via a bottom-up and compositional rule discovery strategy, driven by the semantic seed. The control of the quality of newly acquired knowledge during the bootstrapping process is realized through a ranking and filtering strategy, taking two aspects into account: the domain relevance and the trustworthiness of the origin. A spe-

cial algorithm is developed for the induction and generalization of the ***DARE*** rules. Since ***DARE*** also takes the projections of the target relation and the interaction among these into account, it opens new perspectives for the improvement of recall and reusability of the learned rules.

Various evaluations are conducted that help us obtain insights into the applicability, potential and limitations of the ***DARE*** framework. The comparison of the different data setups such as the size of the semantic seed, the data size and the data source tells us that data properties play an important role in the success of ***DARE***. Furthermore, the evaluation confirms our earlier findings on the influence of proper seed construction for system performance. The detailed qualitative analysis of the ***DARE*** system output encourages us to integrate richer high-quality linguistic processing including discourse analysis.

Zusammenfassung

Informationsextraktion (IE) ist eine Technologie für die Lokalisierung und Klassifikation von relevanten Einzelinformationen in unstrukturierten natürlichsprachlichen Texten und für die Bestimmung der korrekten Relationen zwischen den gefundenen Informationseinheiten. Diese Arbeit ist der Relationsextraktion gewidmet, einer der zentralen Aufgaben der IE. Das Ziel ist, ein generisches Rahmenwerk zu schaffen, das Abbildungen zwischen den linguistischen Analysen und den vorgegebenen semantischen Relationen automatisch lernt. Darüber hinaus soll dieses Rahmenwerk die Anpassung an neue Anwendungsgebiete und neue Relationen mit unterschiedlichen Komplexitäten unterstützen.

Das zentrale Ergebnis dieser Arbeit ist ein neuer Ansatz für die Relationsextraktion, basierend auf einer minimal überwachten Methode für das automatische Lernen der Extraktionsgrammatiken aus einer großen Sammlung von analysierten Texten, das anfangs lediglich durch wenige Beispiele für die gesuchten Relationen gefüttert wird. Diese Startbeispiele werden "semantische Saat" (semantic seed) genannt. Durch den beispielgetriebenen Ansatz kann das System mit minimalem Aufwand an neue Relationstypen und neue Domänen angepasst werden. Es unterstützt Relationen mit unterschiedlicher Stelligkeit und auch deren Projektionen. Ausserdem ist das Rahmenwerk so generisch, dass beliebige linguistische Analysewerkzeug eingesetzt werden können, solange sie die erforderliche Art und Tiefe der Analyse anbieten.

Die Anpassungsfähigkeit und Skalierbarkeit des Rahmenwerks wird durch das ***DARE*** Regelrepräsentationsmodel ermöglicht, das rekursiv und kompositionell ist. Im Vergleich zu anderen IE Regelrepräsentationen, z.B., Stevenson and Greenwood (2006), ist das ***DARE*** Regelrepräsentationsmodel hinreichend ausdrucksmächtig, um eine gute Abdeckung für das Auffinden der Zielrelationen zu gewähren. Die leistungsstarken ***DARE***-Regeln werden durch einen

Konstruktionsmechanismus kompositionell von unten nach oben (bottom-up) aufgebaut. Die Qualitätskontrolle des neu gewonnen Wissens während des "Bootstrapping"–Prozesses wird durch eine Strategie der ständigen Reihung- und Filterung realisiert, die auf zwei Kriterien beruht: der Domänenrelevanz und der Vertrauenswürdigkeit der Herkunft. Ein spezieller Algorithmus wurde für die Induktion und Generalisierung der ***DARE***–Regeln entwickelt. Weil ***DARE*** auch die Projektionen der Zielrelationen und deren Interaktion betrachtet, eröffnen sich neue Perspektiven für die Verbesserung der Trefferquote (*recall*) und für die Wiederverwendkeit der gelernten Regeln.

Unterschiedliche Evaluierungen wurden durchgeführt, um Erkenntnisse über das Anwendungspotenzial und die Beschränkungen des ***DARE***–Ansatzes zu gewinnen. Der Vergleich der verschiedenen Datenparameter, wie Umfang der Beispielmenge sowie Umfang und Herkunft der Lerndaten zeigt deutlich, daß diese Dateneigenschaften ausschlaggebend für den Erfolg des ***DARE***–Einsatzes sind. Darüber hinaus bestätigt die Evaluierung auch unsere früheren Beobachtungen über den Einfluß der Beispielauswahl auf die Systemperformanz. Die ausführliche qualitative Analyse der Ausgaben des ***DARE***–Systems bestärkt uns in der Absicht, in der Zukunft noch tiefere linguistische Verarbeitungskomponenten inklusive einer Diskursanalyse zu integrieren.

Acknowledgments

I wish to thank all my colleagues, friends and relatives who have given me help and encouragement during my work on this dissertation.

Above all, I am deeply grateful to my supervisor Hans Uszkoreit – not only for the numerous fruitful discussions during the development of the thesis. His constant inspiration and his enthusiasm for research encouraged the crystallization of the ***DARE*** idea, filling this intensive time with joy and delight. I have learned a lot from him, especially the courage for new ideas as well as constant skepticism toward both mainstream approaches and one's own research directions and results.

I am especially indebted to Li Hong for her great assistance, particularly in the implementation and evaluation of the ***DARE*** framework. It has been a very pleasant and fruitful cooperation.

It is my honor to have a lineup of internationally renowned scientists in my committee: Matt Crocker, Dietrich Klakow, Doug Appelt, Bill Barry, Erich Steiner and Andreas Eisele. I am specially grateful to Doug Appelt for his valuable suggestions and comments earlier in thesis research and for having agreed to travel a long distance for my defense.

My sincere thanks go to Valia Kordoni and Stephan Busemann for their encouragement, and for their caring criticism now and then, reminding me to focus on this big task in addition to many other interesting projects.

My special thanks to Hans Uszkoreit, Valia Kordoni, Zhang Yi, Rebecca Dridan, Bobbye Pernice and Hans-Urlich Krieger who have helped to proofread the dissertation and generously offered numerous corrections and suggestions.

I am also grateful to Jörg Steffen, Li Hong, Cheng Xiwen and Fu Yu for their great project work and support, allowing me to concentrate on thesis writing during the final phase of this work.

My sincere thanks to Feng Heping, Daniela Kurz, Katja Meder and Brigitte Roth for their care and love in the past years. Special thanks to Selli and his family for their love and support that made me feel at home in Saarland and provided a good environment for my personal development.

A final word of gratitude is dedicated to my beloved husband for his love and support, for accompanying me through late nights during the thesis writing phase, and for preparing delicious meals to keep me strong in the last few months, and also to my parents and my brother Xu Feilong. They were the sources of love and trust from which I have drawn the energy for managing this task and for coping with critical challenges in the last few years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis aims to develop a general framework for the automatic extraction of semantic relations (facts or events) from large collections of natural language texts, a central task of information extraction (IE) research. One of the greatest challenges for IE is to find scalable, adaptive and automatic methods for discovering systematic mappings from general linguistic analyses to different target-specific and unambiguous semantic relations of different complexity. Our proposed solution belongs to the class of minimally supervised machine learning methods, initialized by a small set of samples as representatives of the target semantic relation. The automatic learning method is embedded in a bootstrapping process, a stepwise learning process in which the knowledge acquired at any step serves as the initial knowledge for the subsequent step.

IE has been acknowledged as an urgently needed information technology for the constantly growing digitalized world. The winners in the globalized information society will be people or organizations who can better exploit quick, comprehensive and precise access to digital information for their decision processes than their competitors. Therefore many applications of IE are based on monitoring large dynamic volumes of texts with the aim of detecting relevant pieces of information. Such text collections can be media reports, blogs, corporate websites, patents, technical papers, customer emails, web forums or scientific literature. One useful commercial application is, for instance, the monitoring of customer opinions about products in general and their specific features, which is relevant for product development and marketing strategies. Other applications can be, e.g., the monitoring of innovative technologies and their key players or

the observation of personnel change in a specific sector of industry or trade. Information access would be much easier if all needed sources were structured digital repositories such as traditional data bases. Data stored in this way is easily amenable to semantic search and statistical processing. However, most useful information, in particular, dynamic information, is normally available in unstructured textual formats, e.g., news releases of new products, management succession and political change, customer comments on specific products, and publications of scientific results. Information retrieval technology has made an important contribution to finding documents containing potentially relevant information. However, the relevant pieces of information, i.e., facts, events and opinions, that are contained in the relevant documents are not identified or retrieved directly.

The general task of IE is to extract structured information from unstructured textual data and to link the extracted textual fragments with the original texts. The data format and the semantics of the structured information is defined by the users and the applications. The targeted structured information may be names, concepts or terms belonging to specific semantic classes, or relations among them. Relation extraction is the task of discovering n-tuples of relevant items belonging to an n-ary relation in natural language documents. A theoretically obvious solution is the application of natural language analysis systems for identification of the linguistic units and their relations that correspond to the target semantic structures. In such an arrangement, the IE systems themselves would have to just translate the linguistic roles and relations into the target-specific roles and relations. The more structured the linguistic analysis, the easier the translation step. Yet human language is complex, ambiguous and vague. In order to cope with the complexity, ambiguity and vagueness of language, comprehensive world knowledge would have to be exploited in addition to the results of powerful linguistic analysis. For this reason, semantic interpretation often has to live with underspecified or pseudo analysis, due to the lack of world knowledge and application contexts. Therefore, the ambitious goal of full textual understanding is still far from realistic, if it is purely driven by linguistic motivations. In contrast to the full textual understanding task, IE is only interested in interpreting the textual fragments and their structures that are relevant for the applications. Thus, often only partial textual understanding is needed. Furthermore, the application context is specified explicitly by their users. Each relevant textual fragment should be assigned to a single explicit and unambiguous semantic interpretation. This application-driven

semantic interpretation of natural language texts opens a new perspective of natural language understanding. In this sense, natural language understanding might be regarded as a compositional function of various IE applications in practice (Appelt (2003) and Uszkoreit (2007)). Therefore, it is of theoretical and practical importance to develop a general and adaptable strategy which can identify the relevant linguistic expressions and map their general linguistic analysis to explicit semantic structures automatically defined by different applications.

In the last two decades, IE has developed into one of the most promising and useful applications of natural language technologies. The MUC and ACE programs sponsored by American government institutions (Grishman and Sundheim (1996), (Grishman 1997), Appelt and Israel (1999), Muslea (1999) and Appelt (2003)) have brought researchers together and have accelerated the research process. Among other crucial contributions, the relevant ones are the decomposition of the IE task into several subtasks, the maximized separation of the general linguistic analysis from the domain-dependent analysis and operations, and in particular the development of evaluation standards, gold-standard corpora and the evaluation tool (Hirschman (1998) and Douthat (1998)). Although the specification of the IE task is clear and similar for each application, the solutions can vary depending on the complexity of the tasks and the availability of domain experts and knowledge resources. In practice, the users need an IE system that can quickly adapt to new data and new tasks, but domain experts and high quality domain knowledge, e.g., ontology or textual data annotated with the domain knowledge are in most cases difficult to obtain or their production is connected with high costs. Thus, knowledge-based systems or supervised machine learning methods are only feasible and applicable for certain application scenarios. There is a high demand for methods and strategies that allow an IE system to adapt to new tasks and applications, with minimal human intervention. In recent years, satisfactory results have been achieved for entity recognition and simple binary relation recognition (e.g., Bikel et al. (1999), Brin (1998), Agichtein and Gravano (2000), and Zelenko and Richardella (2003), etc.). The current minimally supervised or unsupervised methods for complex relation extraction such as event extraction cannot demonstrate comparable performance. Some of them apply pattern representation models which have relatively poor expressiveness and thus cannot cover all linguistic constructions representing the target semantic relations (Greenwood and Stevenson 2006). Others try the exhaustive discovery of linguistic patterns without proper fil-

tering and ranking methods, yielding rule sets so large that they destroy the efficiency and even the operability of an IE system (Sudo et al. 2003). Above all, a central problem of most of these pattern learning systems, in particular, the unsupervised systems, is that the learned patterns cannot be employed as relation extraction rules straightforwardly, since the relevant mapping information between linguistic arguments and their semantic interpretation for the target semantic relation is missing.

This thesis proposes a general framework ***DARE*** (Domain Adaptive Relation Extraction based on Seeds). The ***DARE*** framework aims to automatically learn extraction grammars for relations of various complexity from linguistic analysis, taking minimal domain knowledge as input. The mapping between the linguistic arguments and the target semantic arguments is specified automatically. The learning method and its setup is general enough to enable the adaptation to new domains and new tasks.

1.1 Major Contributions

The ***DARE*** framework is highly scalable and adaptable with respect to new domains and relations of different complexity. The scalability and adaptability starts with the decision of taking the relation instances as seed for the bootstrapping-based learning. The relation instances are samples of the target semantic relations defined by the user. Thus, the learning process is driven by the target semantic structures and their complexities. The seed helps us identify the explicit linguistic expressions containing mentions of relation instances or instances of their projections. An interesting study including an empirical investigation analyzes the influence of the seed complexity on the learning performance, considering underspecification and overspecification of the seed semantics. We will give a systematic comparison of the semantic seed-based methods with the methods utilizing the linguistic patterns as seed. Taking the semantic seed as initial input makes the learning system flexible by integrating the suitable linguistic processing components and deciding on the size of the input textual windows for pattern learning.

The scalability and the adaptability of the ***DARE*** framework is mainly supported by its rule representation model. The ***DARE*** rule representation has a high degree of expressiveness, which enables the coverage of all linguistic con-

structions mentioning the relation instances. But the rule productivity (measured by the cardinality of the discovered rules) is comparably low, hence, not a critical influence on system efficiency. The compositional rule representation model enables the construction of pattern rules with various complexities. In the ***DARE*** rule presentation, the linguistic arguments are obligatorily assigned with their semantic roles in the target relation. Parallel to the ***DARE*** rule representation, the ***DARE*** rule extraction algorithm works bottom-up and compositionally: complex rules are built on top of the simple rules for the projections. The rule induction and generalization algorithm also works bottom-up by replacing the specific rules (including those for projections) by more general ones, after the operations of redundancy deletion and clustering.

The ***DARE*** learning process obeys the duality principle introduced by Brin (1998). This means that a good semantic seed helps to find the relevant patterns, and the relevant patterns will extract good semantic seed. At the same time, an inevitable consequence of the bootstrapping design is the effect that newly acquired rules and relation instances potentially contain wrong or noisy information. The ***DARE*** rule ranking and filtering method takes the domain relevance and trustworthiness of origin as its criterion to monitor the quality of the new rules and the new seed.

The ***DARE*** framework is implemented for the English language, utilizing named entity recognition and dependency parsing as its linguistic analysis. Two domains have been selected for our experiments: prize award and management succession. We have chosen prize award as a domain for our experiments because this domain exhibits certain typical properties of application-relevant relation detection tasks. Relations, in particular many complex relations like events, are sparsely represented in large text selections, in our case in freely available news texts. We find the typical skewed frequency distribution of mentions, i.e., some prize events such as Nobel and Pulitzer Prize awards are covered in the text base with great redundancy, many other, less prestigious prizes are mentioned only once or twice. The most prominent prizes give us reliable databases of seeds whereas there are no databases comprising information on all prizes and their recipients. The experiment with the management succession domain using the MUC-6 data provides us an opportunity to compare our method with other minimally supervised pattern learning approaches.

This thesis makes relevant contributions to the evaluation of a minimally super-

vised pattern learning system. It adapts the Ideal Table idea of Agichtein and Gravano (2000) to the Nobel Prize domain to estimate an approximation of the precision and recall value. Furthermore, a systematic evaluation is conducted to investigate the potential and the limitations of the ***DARE*** framework with respect to the number of seeds related to data size. It turns out that the data redundancy plays an important role in the system performance. A more detailed analysis about the interaction with the patterns and the extracted instances helps us gain insights into the crucial properties of more and less suitable domains for the bootstrapping learning approach. In addition, the distribution of rules and instances with respect to their complexity points out the importance of projections for the system performance, in particular, the recall value. Some initial experiments are carried out to test the possibilities for improving the performance for domains exhibiting the typical properties of less suitable data. The results of the error analysis also demonstrate that the precision of the general linguistic analysis has a great impact on the overall system performance.

1.2 Research Context and Support

The thesis idea reported here has undergone its own evolution process accompanied by several research projects that the author has participated in at the Language Technology Lab of the German Research Center for Artificial Intelligence (DFKI)[1].

It started with the WHITEBOARD project, a research grant from the German Federal Ministry of Education and Research (BMBF, FKZ: 01 IW 002). The general goal of WHITEBOARD was to develop a hybrid natural language processing architecture for integrating NLP components of various degrees of depth (Crysmann et al. 2002). On top of the WHITEBOARD hybrid system architecture, an information extraction system architecture emerged in which the relation extraction grammars utilize two different linguistic representation models: regular expressions and predicate argument structures (Xu and Krieger 2003). During this project, the author developed in cooperation with her colleagues a relevant term extraction method via bootstrapping (Xu et al. 2002) and a multilingual shallow information extraction system *SProUT* (Drożdżyński et al. 2004). The research results and insights gained in WHITEBOARD in-

[1] `http://www.dfki.de`

spired the idea development for the ***DARE*** framework. The term extraction tool and the *SProUT* system are applied in the ***DARE*** system.

The project QUETAL[2] was the successor of WHITEBOARD, again funded by the German Ministry for Education and Research (BMBF, FKZ: 01 IW C02). QUETAL was a question answering project in which open-domain and closed-domain question answering techniques were combined in order to improve functionality and performance. The task of the author was focused on the research into a generic strategy of relation extraction from large collections of free texts. Within this project, the author started with the semantic modelling of the prize award domain. The first relation extraction results in the Nobel Prize domain is integrated into a closed-domain question answering system with structured domain knowledge (Frank et al. 2006).

The crystallization of the ***DARE*** idea and its further development (Xu et al. (2006) and Xu et al. (2007)) is partially supported by the HyLaP[3] and RASCALLI[4] projects. HyLaP develops hybrid language processing technologies for a personal associative information access and management application. It is also funded by the German Ministry for Education and Research (BMBF, FKZ: 01 IW F02). RASCALLI is funded by the European Commission Cognitive Systems Programme (IST-27596-2004) and the state of Saarland. During this period, the author supervised a master thesis (Li 2006) which gave an implementation of the system architecture developed in Xu et al. (2006), as initial approach to the ***DARE*** system. Some of the examples and the initial evaluations of the Nobel Prize domain provided in Li (2006) are discussed in this thesis too. The ***DARE*** idea has been applied to the RASCALLI project for the music domain. Felger (2007) is a bachelor thesis supervised by the author, which attempted to apply the learned pattern rules from the Nobel Prize award to the music domain.

1.3 Thesis Structure

The remainder of the thesis is organized in six chapters:

[2] `http://quetal.dfki.de/`
[3] `http://hylap.dfki.de/`
[4] `http://www.ofai.at/rascalli/project/project.html`

Chapter 2 presents background information on information extraction research. It starts with a concise introduction to the definition, the history of IE research, and the relevant parameters for IE system design, followed by a general traditional IE system architecture.

Chapter 3 walks through the approaches and methods that directly inspired the ***DARE*** framework. Three groups of approaches had a strong influence on the ***DARE*** framework: minimally supervised and unsupervised automatic IE pattern extraction methods, research on pattern representation models, and the bottom-up rule induction and generalization strategies. The comparison of the alternative approaches and the insights gained into their problems have helped us in the search for better solutions.

Chapter 4 presents those parts of our own research that inspired, prepared and enabled the ***DARE*** approach and system. Most of these results were obtained in the projects described in Section 1.2.

Chapter 5 provides a detailed representation of the ***DARE*** framework. It explains the basic idea and summarizes the major contributions of the ***DARE*** framework. The following sections provide a detailed description of the system architecture and its key components, the rule representation model, the rule extraction algorithm and the rule induction and generalization method.

Chapter 6 describes the evaluation tasks and reports their results. The evaluation tasks range from a standard precision and recall evaluation to the assessment of detailed system behavior with respect to data properties and system output. An error analysis helps us gain deep insights into the key parameters that influence system performance.

Chapter 7 closes with a conclusion discussing the essential components of our approach. Furthermore, a list of open problems as well as opportunities for future research is presented, classified into three groups: improvement of recall value, boosting precision, and potential applications.

Chapter 2

Information Extraction

This chapter aims to describe the global research context in which our work is embedded. Firstly, we introduce two slightly different definitions of Information Extraction (IE). A survey of the historical development helps us to assess technological progress and scientific insights obtained during the last few decades. We then summarize the relevant components and parameters of an IE system design including document structure, depth of the NLP analysis, complexity of the relation extraction rules, data size, application of statistical and machine learning methods for IE, and evaluation methods. Building on these elements, we will finally describe a generic IE architecture.

2.1 Definition

In general, IE refers to the extraction of relevant information from potentially large volumes of unstructured data. Information can be textual or even multimedia. In this thesis, we select a narrower definition for IE. We regard IE as a pragmatic approach to text understanding (Appelt (2003) and Uszkoreit (2007)). Its task is to gradually approximate the automatic understanding of texts or at least of relevant messages in these texts. IE recognizes the relevant facts or events in texts and identifies their arguments (often entities), ignoring the irrelevant information. The definition of IE on the official NIST web page[1] reads as follows:

[1] `http://www.itl.nist.gov/iad/894.02/related_projects/muc/index.html`

Information Extraction is a technology that is futuristic from the user's point of view in the current information-driven world. Rather than indicating which documents need to be read by a user, it extracts pieces of information that are salient to the user's needs. Links between the extracted information and the original documents are maintained to allow the user to reference context.

The kinds of information that systems extract vary in detail and reliability. For example, named entities such as persons and organizations can be extracted with reliability in the 90th percentile range, but do not provide attributes, facts, or events that those entities have or participate in.

concept	**extracted entities**
prize area	*physics*
person name	*Dr. Robert Laughlin*, *Dr. Horst Stoermer*, *Dr. Daniel Tsui*
monetary amount	*$978,000*
organization	*Stanford University*, *Columbia University*, *Princeton University*

Table 2.1: An example of concept entities

relation	**extracted relation instances**
person, affiliation	
	⟨ *Dr. Robert Laughlin*, *Stanford University* ⟩ ⟨ *Dr. Horst Stoermer*, *Columbia University* ⟩ ⟨ *Dr. Daniel Tsui*, *Princeton University* ⟩
person, prizeArea, monentaryAmount	
person prize area monentary amount	⟨ {*Dr. Robert Laughlin*, *Dr. Horst Stoermer*, *Dr. Daniel Tsui* }, *physics*, *$978,000* ⟩

Table 2.2: An example of relation instances

Thus, the goal of IE systems is to find and link pieces of the relevant information from natural language texts and store these information pieces in a

database format. As an alternative to storing the extracted information pieces in a database, these pieces could also be appropriately annotated in a markup language and thus be made available for indexing and database retrieval. The central IE tasks include finding references to relevant concepts or objects such as names of people, companies and locations, as well as detecting relationships among them, e.g., the birth place of a Nobel Prize winner. Let us look at the following text (2.1) about the Nobel Prize award event:

(2.1) The *Physics prize*, also *$978,000*, will be shared by *Dr. Robert Laughlin* of *Stanford University*, *48*, *Dr. Horst Stoermer*, *49*, a German-born professor who works both at *Columbia University* in *New York* and at *Bell Laboratories* in *Murray Hill*, *N.J.*, and *Dr. Daniel Tsui*, *59*, a Chinese-born professor at *Princeton University*.

If we want to extract events of prize winning, the relevant concepts to be extracted from the above texts are entities such as *prize area*, *monetary amount*, *person name* and *organization* (see examples in Table 2.1). Award relevant relations include the relation between *person* and *organization* and the relation among *person*, *prize area* and *monetary amount* (Table 2.2).

The above NIST definition emphasizes the information discovery aspect of the IE task, while the definition below provided by Wikipedia[2] also explains the applications of IE, e.g., as index for information retrieval, as input for data mining and inference, as markup for data annotation, etc.

> *In natural language processing, information extraction (IE) is a type of information retrieval whose goal is to automatically extract structured information, i.e. categorized and contextually and semantically well-defined data from a certain domain, from unstructured machine-readable documents. An example of information extraction is the extraction of instances of corporate mergers, more formally MergerBetween(company1,company2,date), from an online news sentence such as: "Yesterday, New-York based Foo Inc. announced their acquisition of Bar Corp." A broad goal of IE is to allow computation to be done on the previously unstructured data. A more specific goal is to allow logical reasoning to draw inferences based on the logical content of the input data.*

[2] `http://en.wikipedia.org/wiki/Information_extraction`

The significance of IE is determined by the growing amount of information available in unstructured (i.e. without metadata) form, for instance on the Internet. This knowledge can be made more accessible by means of transformation into relational form, or by marking-up with XML tags. An intelligent agent monitoring a news data feed requires IE to transform unstructured data into something that can be reasoned with ...

2.2 A Brief History

The idea to extract structured information from natural language texts can be found as early as 1987 in the implementation by Sager et al. (1987) of a system for treating medical texts. However, IE as a recognized research area was established several years later by the series of Message Understanding Conferences (MUCs) (Grishman and Sundheim 1996). In the last two decades, IE has grown into a major subfield of natural language processing. The relevant steps in the IE research development are mentioned by various surveys of IE (Grishman and Sundheim (1996), Grishman (1997), Appelt and Israel (1999), Muslea (1999) and Appelt (2003)). Among these steps are the following developments:

- from attempts to use the methods of full text understanding to shallow text processing;
- from pure knowledge-based hand-coded systems to (semi-) automatic systems using machine learning methods;
- from complex domain-dependent event extraction to standardized domain-independent elementary entity identification, simple semantic relation and event extraction.

Thus, IE has evolved into an independent research area with a rich tradition and a broad variety of methods and techniques. In the following, we will present a brief introduction of two important programs which have shaped IE research: Message Understanding Conferences (MUCs) and Automatic Content Extraction program (ACE).

2.2.1 Message Understanding Conferences

MUCs[3] have been organized by NRAD, the RDT&E division of the Naval Command, Control and Ocean Surveilance Center (formerly NOSC, the Naval Ocean Systems Center) with the support of DARPA, the Defense Advanced Research Projects Agency of USA. Grishman and Sundheim (1996) provide a concise overview of the MUCs. MUC is a competition-based conference. It evaluates and publishes the research results contributed by the participants. During the series of the MUCs, the following application domains have been selected:

- MUC-1 (1987), MUC-2 (1989): Naval operations messages.
- MUC-3 (1991), MUC-4 (1992): Terrorism in Latin American countries.
- MUC-5 (1993): Joint ventures and microelectronics domain.
- MUC-6 (1995): News articles on management changes.
- MUC-7 (1998): Satellite launch reports.

The first MUCs started with the ambitious goal of extracting event-oriented n-ary relations, called scenario templates. A template has slots for information about the event, such as the event type, the agent, the time and the location, etc. A template in the MUCs can be very complex, e.g., for MUC-5, the joint venture task requires 11 templates with a total of 47 slots, organized in a hierarchical structure (see a simplified example in Figure 2.2). In order to address the goals of modularity, domain independence, portability and measures of deep understanding, MUC-6 decomposed the IE task into several subtasks, such as named entity recognition, coreference detection, template element extraction and scenario template extraction. MUC-7 has defined the following subtasks as the relevant IE tasks:

- *Named entity recognition (**NE**)*: recognition of entity names (for people and organizations), place names, temporal expressions, and certain types of numerical expressions.

[3] `http://en.wikipedia.org/wiki/Message_Understanding_Conference`

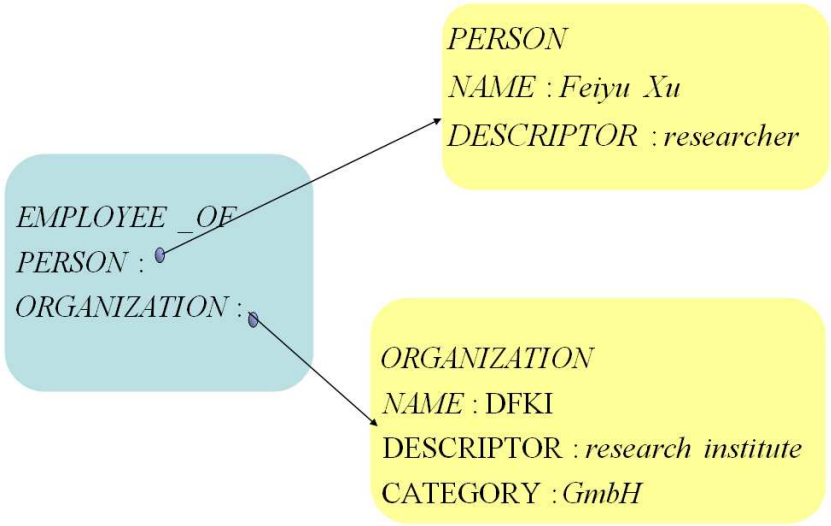

Figure 2.1: Example of template relation

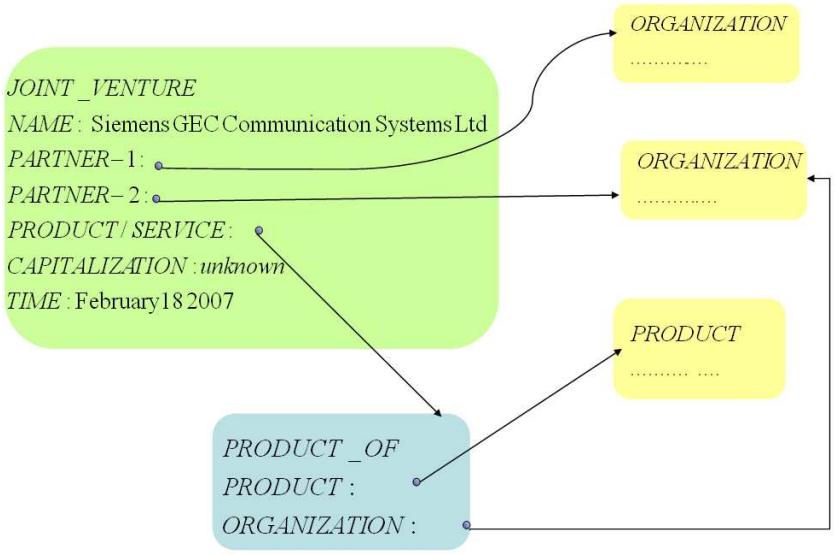

Figure 2.2: Example of scenario template

- *Coreference (**CO**)*: identification chains of noun phrases that refer to the same object. For example, anaphora is a type of coreference.
- *Template element extraction (**TE**)*: filling of small scale templates for specified classes of entities in the texts, where attributes of entities are slot fills (identifying the entities beyond the name level). For example, a person template element contains slots such as name (plus name variants), title, nationality, description as supplied in the text, and subtype.
- *Template relation (**TR**)*: filling a two slot template representing a binary relation with pointers to template elements standing in the relation, which were previously identified in the TE task, e.g., employee_of, product_of, location_of. (see Figure 2.1)
- *Scenario template (**ST**)*: filling a template structure with extracted information involving several relations or events of interest, e.g., identification of partners, products, profits and capitalization of joint ventures (see Figure 2.2).

The participants of each MUC receive descriptions of the scenario along with the annotated *training corpus* in order to adapt their systems to the new scenario. The adaptation duration is from one to six months. After the training phase, they receive a new set of documents (*test corpus*) and apply their systems to extract information from these documents. The results from the test corpus are submitted to the conference organizer, in order to be compared with the manually extracted information (*answer key*). Each subtask has its own answer keys.

The evaluation of the IE systems in MUC was adopted from the information retrieval research community. The precision and recall measures are used for the performance calculation: (2.2) and (2.3). $Number_{correct}$ is the number of correct entities, references or slot fillers found by the system. $Number_{incorrect}$ is the number of incorrect entities, references or slot fillers found by the system. $Number_{key}$ is the number of answer keys, namely, entities, references or slot fillers provided as the gold standard for evaluation.

$$precision = \frac{Number_{correct}}{Number_{correct} + Number_{incorrect}} \tag{2.2}$$

$$recall = \frac{Number_{correct}}{Number_{key}} \tag{2.3}$$

Sometimes an F-measure (2.4) is used as a combined recall-precision score, or to be more precise as the weighted harmonic mean of the two metrics.

$$F = \frac{(\beta^2 + 1) \times precision \times recall}{\beta^2 \times precision + recall} \tag{2.4}$$

The best results reported in MUC-7 (Chinchor 1998) are shown in Table 2.3. Although the named entity recognition has achieved very promising results, the performance of other tasks, in particular the scenario template extraction task, is still very poor.

measure task	NE %	CO %	TE %	TR %	ST %
recall	92	56	87	67	42
precision	95	69	87	86	65

Table 2.3: Best result of MUC-7 for different subtasks

2.2.2 ACE

The MUC was succeeded by a new initiative called ACE, standing for "Automatic Content Extraction"[4]. Its goal was again to stimulate and evaluate progress in IE. The new program followed a pilot study in 1999 (Doddington et al. 2004). ACE aims to develop technologies for partial semantic understanding of texts, including detection and classification of elementary entities, general relations and events explicitly expressed in the texts (Appelt 2003). In comparison to MUC, the types of entities, relations and events are structured ontologically. Furthermore, ACE considers multimedia data using newspaper texts, transcriptions of broadcast data, OCR outputs and blogs. Therefore, the input data often contains poorly-formed texts.

The current major tasks belonging to ACE are:

- Entity detection and tracking (EDT): detects and recognizes all mentions of entities. Entities can be types such as person, organization, geo-

[4] `http://www.nist.gov/speech/tests/ace/`

political, location, facility or their subtypes.

- Relation detection and characterization (*RDC*): detects and recognizes mentions of relations among entities (in particular, entity pairs). There are five general types of relations (role, part, at, near, social) and their subtypes, with a total of 24 types.
- Event detection and characterization (*EDC*): discovers mentions of events where the entities participate. General event types include destroy, create, transfer, move, interact. Event modality is taken into account too.
- Temporal expression detection (*TERN*): requires systems to identify the occurrences of a specified set of temporal expressions and specific attributes about the expressions.

EDT can be regarded as an extension of the MUC NE task, since it not only recognizes the names of entities but also provides their mentions, e.g., pronouns or descriptions. Coreference resolution plays an important role in detecting the equivalent classes of entity mentions. Therefore, EDT is a merging of the NE and CO tasks. ACE has developed a more fine-grained taxonomy of entities than the MUC NE classes. Parallel to EDT, RDC covers the MUC TE and TR tasks, while EDC corresponds to the MUC ST task. However, the event template of EDC is much simpler than the ST task, containing a flat list of arguments with a limited number. The arguments are, for example, agent, object, source, target.

The performance measure for all tasks is formulated in terms of a synthetic application value, in which value is accrued by correctly detecting the target objects and correctly recognizing their attributes, and value is lost by falsely detecting target objects or incorrectly determining attributes of the target object. The overall value of the task performance is the sum of the value for each system output entity (or value, time expression, relation or event), accumulated over all system outputs. The value of a system output is computed by comparing its attributes and associated information with the attributes and associated information of the reference that corresponds to it. Perfect system output performance is achieved when the system output matches the reference without error. The overall score of a system is computed as the system output information relative to this perfect output. The evaluation results of 2007

are published on the NIST website[5]. In Table 2.4, the best results for English documents are summarized.

EDT %	RDC %	EDC %	TERN %
56.3	21.6	13.4	61.6

Table 2.4: Best result of ACE 2007 subtasks for English documents

ACE results cannot be directly compared with the MUC system performance because of the different data setup and evaluation methods.

2.3 IE System Design

There are different parameters which influence a specific IE system design. These are

- document structure of the input texts
  - free text
  - semi-structured
- richness of the natural language processing (NLP)
  - shallow NLP
  - deep NLP
- complexity of the pattern rules for filling templates (so-called template filling rules)
  - single slot
  - multiple slots
- data size of training and application data
- degree of automation
  - supervised
  - semi-supervised
  - unsupervised

[5] http://www.nist.gov/speech/tests/ace/ace07/doc/ace07_eval_official_results_20070402.htm0

- type of evaluation
  - availability of gold standard corpus
  - evaluation measures
  - evaluation of machine learning methods for IE

2.3.1 Document Structure of Input Texts

Typical input texts for an IE system are free texts, which are texts without any meta structure other than use of natural language grammar and punctuation. In our own research field, IE systems only work with free texts, referred to by Muslea (1999) as *IE from free text*. These IE systems generally utilize NLP tools for analysis, forming the traditional IE community.

Parallel to free text IE systems, there are IE systems that extract information from semi-structured texts, such as formatted web pages, building a special area called *information wrapping*. Information wrapping develops techniques which mainly make use of tags in the semi-structured texts as delimiters in their extraction rules. Linguistic structures do not play an important role in the *wrapper* systems.

However, in real world applications, in particular web applications, many information systems combine the two technologies. One simple combination is that in which a wrapper helps to extract free texts from web pages for the IE proper task. Muslea (1999) compiled a survey of differences between linguistically-oriented extraction rules and delimiter-oriented extraction rules. Since our work focuses on free texts, we will not go into any further detail on information wrapping techniques.

2.3.2 IE as Application of NLP

IE is a reasonable application of NLP technologies. NLP tools are often used as preprocessing components for IE systems for identification of domain-independent linguistic structures, ranging from tokens to lexical items, stems, compounds, multi-word terms, phrases, local relationships among phrases, predicate argument structures, sometimes even nested predicate argument structures. The demand for depth of the linguistic analysis is almost parallel to the complexity

of the IE task. In the case of the named entity extraction task, components such as tokenization, morphological analysis, tagging and phrase recognition often provide sufficient structures. The ideal setup for the relation and event extraction tasks would be one in which an NLP system can provide information about dependencies among linguistic chunks (entities), such as grammatical functions or even predicate argument structures. The IE system only has to provide a domain-specific interpretation of the grammatical functions. NLP systems which are designed to deliver such depth of structures are often designed as full text understanding systems, called *deep* NLP systems (Uszkoreit 2002). Although the deep NLP systems tend to deliver more structured and complex linguistic information and have achieved great progress with respect to efficiency and robustness in the last few years, the so-called shallow NLP systems have been preferred by many IE applications when coming to process large amount texts in a limited time, because the shallow NLP systems usually employ efficient local pattern matching techniques (e.g., finite-state techniques) and their analysis results contain very limited ambiguities. Furthermore, the most shallow systems are designed to always deliver analysis results for local textual fragments, thus are robust for real-life applications. The scepticism toward using deep NLP in real-life applications results from their dissatisfactory behavior with respect to efficiency and robustness and also from their inability to deal with the high degree of ambiguity typical for deep NLP.

In a recent development, the demand on high precision information extraction with respect to relation and event extraction is increasing. This requires a deeper and more precise semantic understanding of natural language texts. Some robust semantic-oriented IE systems have emerged (e.g., Surdeanu et al. (2003) and Moschitti and Bejan (2004)). They demonstrate that mapping predicate argument structures or grammatical functions to template structures is more straightforward and efficient than the traditional lexico-syntactic-pattern based approaches (e.g., Hobbs et al. (1997)). At the same time, several attempts (Tsujii (2000), Riezler et al. (2001), Crysmann et al. (2002), Frank et al. (2003), and Xu and Krieger (2003) etc.), have been made to combine shallow and deep NLP, in order to achieve both robustness and precise semantic understanding of free texts. Most of these composition approaches work at the lexical and/or syntactic level, by adding named entity recognition results or chunking results into the deep analysis. The shallow component is responsible for the identification of entities and relationships within a local structure, while the deep component recognizes the linguistic relationships among the entities.

Zhao and Grishman (2005) utilize composite kernels to integrate different levels of linguistic processing including tokenization, sentence parsing and deep dependency analysis. Each level has been trained as a separate kernel. The results show that the composite kernel performs better than a single kernel. HOG (Schäfer 2007) is a further development of the hybrid NLP architecture and provides an infrastructure for extracting information with various complexity. These systems have the advantages of dealing with phenomena where predicate argument relationships are only implicitly expressed in the surface form. Typical examples can be found in linguistic constructions where passive, infinitive VP, control or unbounded dependencies interact with each other.

2.3.3 Template Filling Rules

The complexity of template filling rules plays an important role in the system design when the target relation or event (scenario template) contains multiple arguments. If the template filling rules only fill one argument such as the rules learned by Riloff (1993), it is very difficult for a template merging component to fulfill its task properly because of limited or even missing overlapping information. In general, two partially filled templates can combine with each other, if one subsumes the other, or if there is a coreference link between the arguments (Kehler 1998). Therefore, merging two single argument templates often has to apply a less reliable but pragmatic heuristics, namely, the closeness between the two textual segments from which the two templates are extracted.

Muslea (1999) presents a set of systems that learn multi-slot template filling rules from annotated corpora. These corpora are usually analyzed by a sentence parser. Given the linguistic structures and their associations with the target template arguments, template filling rules define the corresponding mapping between linguistic arguments and the template arguments. The following example is a very simple two-slot template filling rule for the management succession domain.

(2.5) $\langle$ *subject: personIn* $\rangle$ succeeded $\langle$ *object: personOut* $\rangle$

Linguistic structures such as grammatical functions provided by deep NLP provide better inputs for multi-slot template filling rules than structures delivered by shallow NLP, because deep linguistic structures are not restricted to the

local textual fragments where usually fewer arguments can be embedded in.

2.3.4 Data Size

The ultimate goal of IE is to discover information in an enormous volume of texts within a realistic time limit. Google as the currently most successful search engine confirms that information retrieval (IR) (Salton and McGill 1986) is able to find relevant information in real time from a large amount of data. However, Google results are lists of relevant documents instead of structured data records.

In comparison to IR, IE needs more CPU power for text analysis and other operations. Therefore, an IE system has to find a suitable tradeoff between data size, analysis depth, complexity of the target structures and time constraints. Deeper analysis and extraction of more complex template structures consumes more time than shallow analysis and simple named entity recognition or binary relation extraction.

One very promising application area of IE is question answering (Voorhees 2003). Many question answering systems (e.g., Harabagiu et al. (2000), Voorhees (2003), Neumann and Xu (2003) and Harabagiu et al. (2003)) utilize IR for the detection of relevant documents or paragraphs from a large amount of data and apply IE only to extract more structured information from the selected texts.

2.3.5 Automatic Knowledge Acquisition

The high demand for IE systems that are portable to new tasks and domains pushes the development of automatic methods that can acquire knowledge at various levels for new applications and new domains without the use of human experts.

In the last few years, extensive research has been dedicated to entity recognition and simple relation recognition with quite significant results (e.g., Bikel et al. (1999) and Zelenko et al. 2003; etc.). A particularly important task is the acquisition of scenario pattern rules. The machine learning approaches to acquiring pattern rules can be grouped into supervised, minimally supervised and unsupervised methods (e.g., Riloff (1993), Riloff (1996), Califf and Mooney (1999), Brin (1998), Agichtein and Gravano (2000), Yangarber (2001), Green-

wood and Stevenson (2006), Suchanek et al. (2006), Sudo et al. (2003) and Davidov et al. (2007)).

Supervised methods assume a corpus of documents annotated with the slot filler information. Therefore, they are often faced with the problem of missing high quality corpora for new domains. Muslea (1999) gives a survey of the supervised pattern acquisition methods developed by systems such as AutoSlog (Riloff 1993), LIEP (Huffman 1996), PALKA (Kim and Moldovan 1995) and RAPIER (Califf and Mooney 1999). All these systems are dependent on a well-annotated corpus with an adequate data property. This means that the data is assumed to provide a broad coverage of examples and possesses at the same time sufficient data redundancy.

Minimally supervised learning seems a very promising approach. This learning method acquires knowledge automatically and is initialized by a small set of domain knowledge. Systems such as DIPRE (Brin 1998), Snowball (Agichtein and Gravano 2000) and ExDisco (Yangarber 2001) take a small set of domain-specific examples as seed and an unannotated corpus as input. The seed examples can be either target relation instances or sample linguistic patterns in which the linguistic arguments correspond to the target relation arguments. New instances or new patterns will be found in the documents where the seed is located. The new instances or patterns will be used as new seed for the next iteration. The whole iteration process is referred to as *bootstrapping* (Abney 2002).

The unsupervised systems do not make use of any domain-specific information. Systems like Sudo et al. (2003), Turney (2006) and Davidov et al. (2007) attempt to detect patterns where the relevant entities or concepts are located. However, these pattern rules can only be employed as the trigger parts of the relation extraction rules, because the mappings between the linguistic arguments and domain-specific semantic filler roles are missing.

2.3.6 Evaluation Methods

A crucial contribution of the MUC conferences to IE research is the development of the evaluation methods, standards, data and tools (Hirschman (1998) and Douthat (1998)). The precision and recall measures introduced by the MUC

conferences have become widely accepted as the evaluation standard for the performance assessment of most IE systems. As explained in section 2.2.1, the systems precision was defined as the number of slots filled correctly, divided by the number of slots actually filled. Recall was defined as the number of slots it filled correctly, divided by the number of possible correct fillers specified in the gold standard corpus. F-measure is a weighted combination of precision and recall for providing a single value of the system performance.

There are also other corpora such as the job postings collection (Califf 1998), and seminar announcements, corporate acquisition and university web page collections (Freitag 2000) published for the research community. In the research community, many systems apply their methods to these corpora for purposes of comparison.

Agichtein and Gravano (2000) provide a pragmatic method for dealing with evaluation of IE systems without annotated corpora. They make use of a publicly available structured database which covers a large list of relation instances of the target relation. Given such an external structured database, they compile a so-called *Ideal* table from the textual collection, to which the IE system applies. They detect all instances from the database mentioned in the textual collection. They are not interested in all mentions of the relation instances. If one mention of a relation instance is detected, the system is then successful for this relation instance. Precision and recall values can be computed based on this ideal table. The method is feasible when some external almost complete gold standard database for the target relation is available.

However, the availability of suitable corpora for different applications and different methods is still an unsolved problem. The data properties of the available corpora such as the MUC corpora are often too small and restricted to newspaper texts. Statistical and machine learning methods that rely on large amounts of data and data redundancy cannot properly be trained and evaluated by these corpora. Daelemans and Hoste (2002) pointed out similar problems for the evaluation of machine learning methods for other NLP tasks, namely, the lack of consideration of interaction between data property, information source (the variety of the data processing features, e.g., application of shallow vs. deep NLP) and the method or algorithm parameter setting. The above problem has also been mentioned by a survey paper of the IE evaluation tasks by Lavelli et al. (2004). They also discuss the problems of assessment of inexact identification

of filler boundaries and the possibility of multiple fillers for a slot and potential solutions to these two problems. They require that an IE task specifies the following three issues:

- a set of fields to extract
- the legal number of fillers for each slot: "exactly one", "zero or one", "zero or more" or "one or more values"
- the possibility of multiple varying occurrences of any particular filler

Concerning machine learning methods, it is agreed that only precision and recall values alone are not informative enough to explain the system performance (Lavelli et al. (2004) and Ireson et al. (2005)). The analysis of the learning behavior with respect to the learning curve is very important in understanding the system.

2.4 A Generic and Traditional IE Architecture

A generic IE architecture emerged during the MUC period (Appelt and Israel 1999). This architecture applies shallow text processing methods and solves the subtasks in a cascaded sequential workflow (see Figure 2.3). The architecture illustrated in Figure 2.3 is a slightly modified version of the architecture depicted by Appelt and Israel (1999). Many IE research groups have developed their systems based on this system design, using finite-state technologies, e.g., the pioneering system FASTUS (Hobbs et al. 1997), SMES (Neumann et al. 1997), GATE (Cunningham 2002) and *SProUT* (Drożdżyński et al. 2004). These systems use shallow text understanding technologies (local pattern matching) to cope with the problems in efficiency and robustness found in the traditional full text understanding systems.

The components in the architecture can be classified in two groups: local text analysis and discourse analysis. The local text analysis components are responsible for recognition and classification of the linguistic and domain-specific functions of words or phrases and their linguistic and domain-specific relations within a sentence boundary. The discourse analysis tries to detect relationships among the domain relevant linguistic objects beyond the sentence boundary,

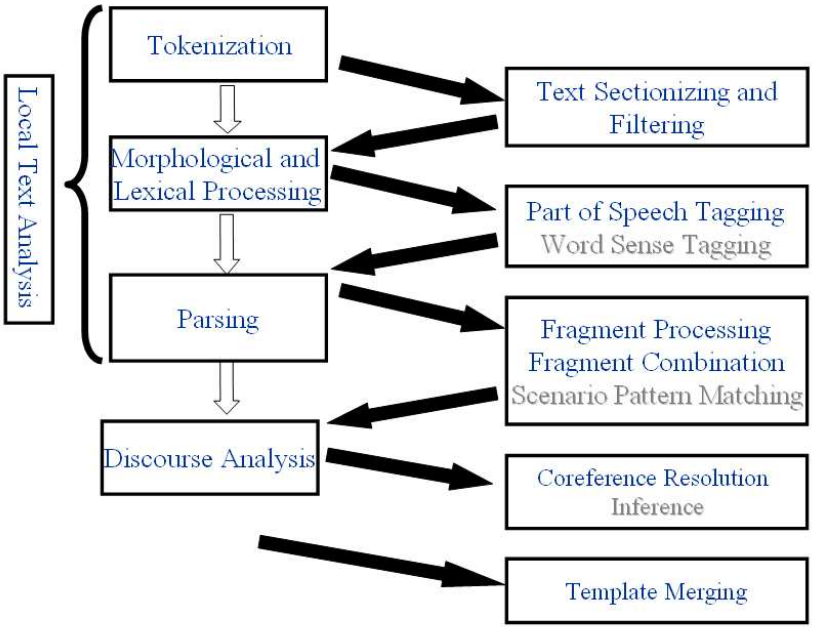

Figure 2.3: traditional IE architecture

e.g., coreferential, temporal, causal. As depicted in Figure 2.3, these components are

- **Local text analysis**
  - Tokenization
  - Morphological and lexical processing
  - Parsing
- **Discourse analysis**

The grey colored subcomponents are domain-specific.

Most ***tokenization*** tools are responsible for detection of word, clause and sentence boundaries. Some extended tools also classify the words to token types based on their internal string structures, e.g., two digits, lower case or capitalized word. The classification information is often used by named entity recognition. For European languages, white space is a good indicator of word

boundaries and punctuation is often taken into account for the recognition of clause and sentence boundaries. In case of Chinese and Japanese texts where words are not separated by white spaces, language specific word segmentation tools are developed for each language. For some special applications, preprocessing tools for *text sectionizing and filtering* are needed in order to detect the document structure and identify the relevant document parts.

The ***morphological and lexical processing*** component has to deliver the linguistic analysis of single words, e.g., word class, morphological functions. The complexity of the morphological and lexical analysis differs from one language to another. For example, English has a very simple inflectional morphology, while a German IE system often demands compound analysis. *Part of speech tagging* and *word sense disambiguation* are useful for selecting one word class and one word sense among the different readings.

The task of ***parsing*** in such an IE architecture is to i) recognize linguistic units and their relations; ii) classify the linguistic functions and domain-specific functions of the linguistic units; iii) recognize the domain-specific relations on top of the linguistic relations. Three core components are *fragment processing*, *fragment combination* and *scenario pattern matching*. In many systems, *the fragment processing* is realized as *named entity recognition* and *phrase recognition*. Named entity recognition is the core task in MUCs and the ACE program. The phrase recognition detects noun phrases, verb phrases, prepositional phrases, etc. The *fragment combination* often refers to partial or even full sentence analysis, where relations among phrases within a sentence are constructed, e.g., verb phrase with modification, noun phrase with a prepositional phrase as its attachment, phrase structure of a clause, dependency structure of a sentence. The *scenario pattern matching* deals with recognition of the domain specific relationships among the constituents and assigns the domain-specific argument roles to the constituents, utilizing the linguistic relations as indications. At this stage, each pattern matching yields a filled subset of the arguments in a scenario template.

Often the arguments that can fill a scenario template are not located in a single sentence and are scattered throughout the whole document. For example, the date time description of an event can be mentioned at the beginning of a newspaper report and is referred to later in the document. Therefore, *coreference resolution* plays an important role in ***discourse analysis***, in order to

find the links among the references and to build equivalence classes of entities mentioned in the document. The *inference* component helps to derive facts explicitly from implicit linguistic expressions and existing facts, e.g., *X succeeded Y* means in the management succession domain that X fills the role of starting the job (personIn) and Y fills the role of leaving the job (personOut). The *template merging* component attempts to combine the partially filled templates to one scenario template, with the help of the facts derived by inference and the resolved coreference information. The coreference information is often a good indicator to show that two partially filled templates refer to the same relation or event.

The core components in this generic architecture can be found in both rule-based and statistical IE systems. In the early systems at MUCs, the knowledge and rules were mostly constructed manually by experts. A typical knowledge-based rule system is FASTUS (Hobbs et al. 1997). The change of application domains for each MUC encourages portability of systems. This data setup situation simulates the first learning approaches such as those of Riloff (1993) and Riloff (1996)) to training automatic systems. Furthermore, the development of domain-independent and reusable components, e.g., named entity recognition, coreference resolution, simple relation recognition is also very important for system portability. The ACE program pushes the research in this direction.

2.5 Conclusion

This chapter has attempted to give a concise overview of the IE research area. The organized competitions such as MUCs and ACE have provided the research community with opportunities and infrastructures for coming together and comparing methods. These events have made valuable contributions to progress in this area and have, at the same time, presented good examples for other disciplines too. Although the IE task specification is clear, the solutions to the core problems can vary in different ways. They range from knowledge-based, to statistical and machine learning methods, or even to their combination. The above sections illustrate that research from different disciplines influences the IE research development. These include NLP, IR, knowledge engineering, machine learning, web technologies, question answering, etc. The list of relevant parameters and their interaction for the IE system design present the problems

and challenges in this area. Although this area is becoming more and more complex and methods are becoming more sophiscated, IE is one of the most promising and useful applications of NLP.

Chapter 3

State of the Art

In this chapter, we will give a detailed description of the relevant related work that has provided us with useful methodology, positive or negative evidence in favor of certain decisions or other forms of scientific inspiration. We group the related work into three areas:

- Minimally supervised and unsupervised automatic IE pattern extraction methods
- Research on pattern representation models
- Bottom-up rule induction and generalization

In the following, we will introduce each approach and discuss its advantages and problems.

3.1 Minimally Supervised and Unsupervised ML Methods

The motivation behind minimally supervised and unsupervised machine learning (ML) methods is the goal of acquiring IE patterns with minimal human intervention. AutoSlog-TS (Riloff 1996) is the first system which only uses a pre-classified unannotated text corpus. It extracts linguistic patterns that are instantiated with domain relevant lexical trigger words. The DIPRE system

(Brin 1998) introduces a method for learning pattern rules from a large volume of web data, taking a very limited set of relation instances as initial knowledge. The data is not classified in advance. The whole process runs in a bootstrapping manner. The pattern rules are composed of HTML tags and slot fillers. Following the DIPRE system, many derivative and alternative approaches for IE pattern learning have emerged (e.g., Sudo et al. (2001), Pantel and Pennacchiotti (2006), Greenwood and Stevenson (2006), Blohm and Cimiano (2007)). The Snowball system series (Agichtein and Gravano (2000) and Agichtein et al. (2000)) and the ExDisco system (Yangarber 2001) demonstrate the most influential approaches of this type. Our method is built on top of their core ideas.

3.1.1 AutoSlog-TS

AutoSlog-TS (Riloff 1996) takes pre-classified texts as a training corpus, namely, *relevant* and *irrelevant* documents. The pattern acquisition process contains two stages:

1. ***pattern extraction***: the sentence analyzer produces a syntactic analysis for each sentence and identifies noun phrases. For each noun phrase, the heuristic rules generate a pattern to extract a noun phrase, for example,

   <subject> bombed.

2. ***relevance filtering***: the entire text corpus is processed for a second time using the extracted patterns obtained by stage 1. Then each pattern will be assigned a relevance rating based on its occurrence frequency in the relevant documents relative to its occurrence in the total corpus. A preferred pattern is one that occurs more often in the relevant documents.

AutoSlog-TS uses 1500 MUC-4 development texts, of which about 50% are relevant. In stage 1, AutoSlog-TS generates 32,345 unique extraction patterns. After discarding patterns with frequency "1", 11,225 remain. The remaining patterns are ranked based on the relevance filtering function. A user reviewed the top 1970 patterns in about 85 minutes and kept the best 210 patterns.

In addition, the user labelled the noun phrase slots with the corresponding template roles.

After evaluation of the supervised learning system AutoSlog (Riloff 1993) and the AutoSlog-TS system against the same test corpus, it turns out that AutoSlog-TS returns a comparable performance. The advantage of AutoSlog-TS in comparison to the supervised approaches is that it needs much less manual annotation effort. It is rightly viewed as one of the pioneering approaches to automatic learning patterns without annotation. However, the learned patterns need domain expert knowledge for assigning semantic roles to the linguistic arguments. Furthermore, the ranking function is not optimal, because it is too dependent on the occurrence of the pattern. Hence, relevant patterns with lower frequency will not float to the top.

3.1.2 DIPRE – Dual Iterative Pattern Relation Expansion

Brin (1998)'s DIPRE system uses a bootstrapping method to find patterns without any pre-annotation of the data. The process is initiated with a seed set of pairs in some given relation, such as author–title. In his experiment, five author–title pairs are selected. The system then searches a large corpus for patterns in which one of these pairs appears. Given these patterns, it can then find additional examples and add them to the seed set. The process can then be repeated. This approach takes advantage of facts or events which are stated in multiple forms within a corpus.

The algorithm is rather straightforward:

- ***input***: web pages with urls and a small set of relation instances. 24 million web pages in `http://google.stanford.edu` with 147 gigabytes belong to the extraction corpus.
- ***steps***
  - ***occurrence identification***: find all occurrences of the relation instances in the corpus
  - ***pattern extraction***: generation of patterns based on found occurrences

    - group occurrences according to their *order* and *middle*. *order* specifies the linear precedence between the two arguments. *middle* is the HTML tag structure between them;
    - for each group, generate a pattern obeying the specificity constraint
    - associate the url pattern with the text pattern, e.g.,
        - **url-pattern**: $www.sff.netlocusc.\star$
        - **text-pattern**: $< LI >< B > title < B > by\ author\ ($

- ***pattern application***: apply patterns to training corpus to obtain additional relation instances
- use the expanded relation seed for the next iteration

Brin makes a very important contribution to research work in the seed-based IE pattern learning. He introduces a duality principle which drives the bootstrapping process. The underlying insight is: good seed samples lead to good patterns, while good patterns help to extract good instances. Good patterns are patterns that have high coverage (high recall) and low error rate (high precision). Good instances are instances that are realized by good patterns. In his experiment, among the five examples, only two of them have led to patterns for further extraction. Brin warned of the error spreading potential in a bootstrapping process, since any noisy or wrong information can hurt the performance dramatically when applying it to a large amount of data in the further iterations. Therefore, Brin discussed this danger and developed initial suggestions for rule scoring and filtering. However, these methods are still based on simple heuristics. Because of the missing annotated data, the evaluation was only carried out on small samples.

3.1.3 Snowball System: Relation Extraction from Plain Texts

Agichtein and Gravano (2000) present the *Snowball* system which extracts relations from large plain texts without HTML tags. The plain texts are annotated with recognized named-entities such as *company* and *location*. Snowball employs a kind of bootstrapping method which learns patterns from existing relation instances and extracts new relations from learned patterns iteratively. The initial run is supported by a seed of example relation instances. Snowball considers only one binary relation in the experiment, namely, the *location* of the

headquarters of a *company*. Snowball can be regarded as a further development of the DIPRE approach. The contributions of Snowball include:

- **techniques for generating patterns and extracting tuples**:

  A pattern in Snowball is presented as a 5-tuple

  $$< left, tag1, middle, tag2, right >,$$

  where $tag1$ and $tag2$ are named-entity tags, and $left$, $middle$, and $right$ are vectors associating weights with terms.

  This representation is used for both pattern generation and relation extraction. The pattern generation uses a simple single-pass clustering method to group similar tuples and generate a corresponding new pattern. New relation mentions are identified via the match between the 5-tuple representation of a candidate text fragment with the 5-tuple representation of the pattern. A candidate text fragment is a piece of text in a sentence including the relation relevant named entity pairs.

- **strategies for evaluating patterns and relation instances**:

  During each bootstrapping iteration, Snowball evaluates the confidence of patterns and extracted relation instances to obtain high quality patterns and reliable relation instances. The confidence of a pattern $Conf(P)$ depends on the precision of its extracted relation tuples:

  $$Conf(P) = \frac{P.positive}{P.positive + P.negative} \tag{3.1}$$

  where P.positive is the number of positive matches for P and P.negative is the number of negative matches.

  The confidence of a relation instance can be calculated from the confidence of the patterns which extract this relation and the similarity between matched text fragment and the pattern:

$$Conf(T) = 1 - \prod_{i=0}^{|p|}(1 - (Conf(P_i).Match(C_i, P_i))) \tag{3.2}$$

where $P = \{P_i\}$ *is the set of patterns that generated* T *and* C_i *is the context associated with an occurrence of* T *that matched* P_i *with degree of match* $Match(C_i, P_i)$.

- **evaluation methodology and metrics**:

  In order to cope with the missing gold-standard corpus for evaluation, Snowball has adapted the precision and recall metrics from information retrieval to quantify how accurate and comprehensive the table of the extracted tuples is. They propose an *Ideal* table. The table uses the headquarter lists of companies given by an existing database, namely, Hoover's compiled table on the Web, and selects only the location and organization pairs mentioned in the text collection. Since the extracted tuples also contain instances beyond the Ideal table entries, it compiles a *join* table. *Join* table contains organizations occurring in the *Ideal* table as well as in the extracted table. The precision and recall value is then defined as follows:

$$Recall = \frac{\sum_{i=0}^{|Join|}([locationInExtracted_i = locationInIdeal_i])}{|Ideal|}.100\% \tag{3.3}$$

where $[locationInExtracted_i = locationInIdeal_i]$ *is equal to 1 if two locations match each other.*

$$Precision = \frac{\sum_{i=0}^{|Join|}([locationInExtracted_i = locationInIdeal_i])}{|Join|}.100\% \tag{3.4}$$

In comparison to traditional information extraction systems, Snowball does not attempt to capture every instance of a tuple. Instead, it is successful for a relation tuple if one of its instances in the document collection is discovered.

Snowball uses large collections of newspapers from the North American News Text Corpus, available from LDC. The *training* collection consists of 178,000 documents, all from 1996, while the *test* collection is composed of 142,000 documents, all from 1995 and 1997. The performance of Snowball is described via the following evaluation results:

- 96% precision (manually computed precision estimate, derived from a random sample of 100 tuples)
- The evaluation against the *Ideal* table results in a precision of 69% and a recall of 75% when all tuples in Ideal are equally considered.

A re-implementation of the DIPRE system is carried out for the evaluation by Agichtein and Gravano (2000). The strategies for evaluating patterns and relation instances after each iteration give rise to better performance of the Snowball system in comparison to DIPRE after the first run. DIPRE cannot avoid producing noisy and wrong patterns and instances in the further iterations because it does not have a good mechanism to control the quality of patterns and new seeds. Snowball can set a threshold for each iteration and make it adaptable to the precision or the recall requirements of each application.

The Snowball system makes very important contributions to the seed-based pattern learning methods, in particular, the scoring and filtering strategies of patterns and instances, and the novel evaluation methods. However, its pattern representation is too much based on the surface strings. Therefore, too many specific rules have to be produced in order to cover most linguistic expressions. Furthermore, surface-oriented pattern representations are unsuitable for recognizing relationships expressed via nonlocal linguistic constructions.

3.1.4 ExDisco: Automatic Pattern Discovery

A major milestone in the development of IE pattern-learning is Yangarbers's ExDisco system, described in great detail by Yangarber (2001). The system incrementally learns domain relevant patterns from un-annotated but parsed free texts, starting with a small set of *pattern samples* as seed.

The original goal of ExDisco is to learn patterns that are suitable for extracting complex relations or events at the scenario template level. However, its patterns

are restricted to the *subject-verb-object* constructions. Therefore, they are only able to extract unary and binary relations. In comparison to the DIPRE and the Snowball systems, the ExDisco system only focuses on pattern extraction and document classification. The whole bootstrapping process is composed of iterations for pattern extraction or document classification: documents are classified as relevant and irrelevant according to the occurrences of the seed patterns, while relevant patterns are extracted from the relevant documents. The relation extraction is not integrated in the system architecture. Yangarber (2001) makes two fundamental assumptions for his method:

- **principle of density**: relevant texts contain more relevant patterns
- **principle of duality**:
  - documents that are relevant to the scenario are strong indicators of good patterns
  - good patterns are indicators of relevant documents

The duality principle is analogous to that defined in the DIPRE system, the only difference being that ExDisco considers the relationship between patterns and documents instead of that between patterns and instances.

This is the main algorithm of ExDisco:

- ***input***:
  - **(a)** a large corpus of un-annotated and un-classified documents
  - **(b)** a trusted set of scenario patterns, initially chosen ad hoc by the user as seeds. Normally the seed is relatively small, containing two or three samples.
  - **(c)** (possibly empty) set of concept classes: e.g., person, company, position
- ***document classification***: apply seeds to the documents and divide them into relevant and irrelevant groups
- ***pattern extraction***:
  - automatically convert each sentence into a set of candidate patterns.

  - choose those patterns which are strongly distributed in the relevant documents. Special measures are defined for scoring the pattern relevance and the document relevance, in order to control the quality of the new seed.

- ***user feedback***

- ***repeat***: until no more patterns can be discovered.

ExDisco is an NLP-based IE pattern learning system. It utilizes named entity recognization for text normalization. Furthermore, it applies a general-purpose dependency parser of English, based on the FDG formalism (Tapanainen and Jarvinen 1997) and maps natural language sentences and clauses to subject-verb-object constructions. Finally, an inverted index of subject-verb-object tuples is produced for the entire corpus.

Three sorts of evaluation have been conducted in this work:

1. ***qualitative evaluation***: manually inspecting the extracted patterns
2. ***text filtering***: testing the recall and the precision of the ExDisco-classifier as an IR document retrieval system
3. ***event extraction***: integrating the extracted patterns into an existing IE system and testing recall and precision

pattern base	recall %	precision %	F %
seed	27	74	39.58
ExDisco	52	72	60.16
union	57	73	63.56
manual-system	47	70	56.40
union	56	75	64.04

Table 3.1: Evaluation of event extraction: test data of management succession

The third evaluation method is the most interesting one for the IE task. In Table 3.1, Yangarber (2001) shows the performance of the seed, the learned new patterns and their union, as well as the influence of the learned patterns on the performance of an existing system. Since the patterns learned by ExDisco are not labelled with the semantic roles, they are all manually augmented with their slot filler roles before being integrated into the existing IE system. The following

example (3.5) is an ExDisco pattern, where the semantic role of the object argument *person* is underspecified. In the management succession domain, the person in this pattern plays the *PersonIn* role, i.e., somebody obtaining a new position.

(3.5) ⟨subject: company⟩ verb:"appoint" ⟨object: person⟩

The above table also shows that the ExDisco system not only delivers better performance with its learned patterns than the manual system and but that it also improves the performance of the existing manual system.

However, as observed by Yangarber (2001) himself, the patterns learned here are incomplete relation extraction rules. They only contain the trigger part. Just as in most other automatic pattern discovery systems, information about slot filler labelling is missing in these learned rules. Moreover, the expressiveness of the ExDisco rule representation is still very limited. The subject-verb-object construction only covers a subset of all linguistic expressions representing the potential relation and event instances in texts.

3.2 Pattern Representation Models

The existing minimally supervised and unsupervised automatic approaches learn patterns from document structures, such as HTML tags like the DIPRE system (Brin 1998) or linguistic annotations such as named entity tags (Agichtein and Gravano 2000), deeper linguistic analysis such as grammatical relations or even their combination (Yangarber 2001). The Snowball system makes use of named entity tags, surface strings and their linear order as components in its pattern representation. This pattern representation is applicable to a binary relation such as the headquarter's location of a company, because variants of linguistic expressions for this kind of binary relations are very limited and the slot fillers often co-occur in local linguistic structures. However, it is difficult to adapt this pattern representation to

- scenario-level relations or events where multiple slot fillers (in general more than two) are involved. The slot fillers are not only expressed within local consecutive text fragments.

- languages with rich morphology, rich grammatical constructions or free word order (e.g., German and Japanese).

In addition, patterns bound to surface strings and surface linear order are too close to the training data and are often not applicable to unseen data. Therefore, many approaches, targeted to extracting complex relations or events, develop their pattern representations on top of a dependency analysis (e.g., Yangarber (2001), Sudo et al. (2001) and Greenwood and Stevenson (2006)). Sudo et al. (2001) point out that the subject-verb-object (SVO) constructions proposed by the ExDisco system are not expressive enough to cover complex linguistic patterns (e.g., verb chains) and that they are often too general, yielding bad precision. They suggest a chain model as pattern representation. A chain is a path in a dependency tree, dominated by a verb. Although a chain provides potentially more contextual information for an argument, information about the relations among arguments in different paths is lost. Hence, the expressiveness of this model is limited too. As an improvement, Sudo et al. (2003) suggest a subtree model which combines the expressiveness of the SVO constructions and the chain model. The subtree model treats all subtrees and paths dominated by verbs in a dependency tree as its patterns. However, the computational burden caused by the large number of subtrees for further rule filtering and rule induction is quite heavy. Greenwood et al. (2005) propose a linked chain model that allows the extraction of pairs of chains in addition to single paths. This can be regarded as a simplified compromise between the chain and the subtree model.

Stevenson and Greenwood (2006) present a systematic comparison of the various pattern representation models. Given a dependency analysis example by them depicted in Figure 3.1, each model produces different number of patterns with different complexity.

The SVO construction extracts two patterns from the dependency tree. They are

(3.6) (1) [V/hire] (subj[N/Acme Inc.] + obj[N/Mr Smith])
(2) [V/replace] (obj[N/Mr Bloggs])

The chain model extracts eight patterns. Some of these are listed below.

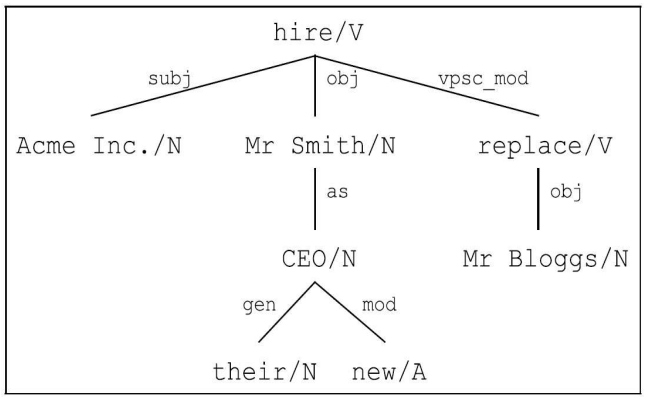

Figure 3.1: Dependency structure analysis

(3.7) (1) [V/hire] (subj[N/Acme Inc.])
(2) [V/hire] (obj[N/Mr Smith])
(3) [V/hire] (obj[N/Mr Smith] (as[N/CEO]))
(4) [V/hire] (vpsc_mod[V/replace] (obj[N/Mr Bloggs]))
(5) [V/replace] (obj[N/Mr Bloggs])

The linked chain model extracts 14 patterns, see some of the linked chains below.

(3.8) (1) [V/hire] (subj[N/Acme Inc.] + obj[N/Mr Smith])
(2) [V/hire] (subj[N/Acme Inc.] + obj[N/Mr Smith] (as[N/CEO]))
(3) [V/hire](obj[N/Mr Smith] + vpsc_mod[V/replace] (obj[N/Mr Bloggs]))

The subtree model extracts a superset containing the patterns from the above models and also other subtrees. For this example tree, it can derive 42 patterns.

Stevenson and Greenwood (2006) present a formal calculation to enumerate the number of patterns produced by each model (Table 3.2). In their experiments, they count the number of patterns each model produces with respect to three different dependency parsers. They take MUC–6 management succession corpus (MUC-6 1995) and other corpora for experiments. Three dependency parsers are MINIPAR (Lin 1998), the Machinese Syntax parser from Connexor Oy

(Tapanainen and Jarvinen 1997) and the Stanford parser (Klein and Manning 2003). The following table gives an overview of the number of patterns in each model.

Parser	SVO	Chains	Linked chains	Subtrees
MINIPAR	2,980	52,659	149,504	353,778,240,702,149,000
Machinese Syntax	2,382	67,690	265,631	4,641,825,924
Stanford	2,950	76,620	478,643	1,696,259,251,073

Table 3.2: Number of patterns produced for each pattern model by different parsers for MUC-6

Although the number of the patterns produced by MINIPAR for SVO, chains and the linked chains is comparable with other parsers and is even smaller with respect to the chain and linked chain models, the number of the subtrees is several orders of magnitude higher than the others. The reason is that MINIPAR provides a special treatment of linguistic phenomena such as conjunction, anaphora and VP-coordination, where the same grammatical function can be shared by different heads. This leads to duplication of tree structures when coming to extract subtrees. For example,

(3.9) Peter is jumping and dancing.

The person name "Peter" is a subject of two different verbs "jump" and "dance". Each verb dominates a separate subtree.

A further experiment in this work is to measure the coverage of the patterns with respect to the events mentioned in the corpus. It turns out that the SVO representation has the lowest coverage, i.e., 6% average of all corpora for any parser. This indicates that SVO representation is not expressive enough for the IE task. The chain model has achieved around 40%, still relatively low. The linked chain model covers almost 95% of all relation instances by using the Stanford parser, the highest coverage among the three parsers. The highest coverage is reached by the subtree model. However, the subtree model is not suited for inducing a proper rule set in a realistic time limit, which is a central prerequisite for the portability of any IE system.

Patterns acquired by all four models only contain the trigger part of the relation extraction rules, and the mapping between the linguistic arguments and the relation-specific semantic roles is unspecified.

3.3 Rule Induction and Generalization

In the last sections, we present methods for learning pattern rules from analyzed natural language sentences. All these methods follow a bottom-up rule learning strategy. They learn highly specific rules from corpus examples. The number of pattern rules is a relevant factor for the efficiency and effectiveness of an IE system. It is infeasible for an IE system to cope with a large number of specific rules as produced by the subtree model. Therefore, rule generalization is an essential task for keeping the rule set at a computable size and to provide for efficient rule search and matching. Furthermore, general rules are more adaptable to unseen data than specific rules. Therefore, a good pattern learning system tries to minimize the rule set by generalizing while avoiding any overgeneralization. The tradeoff between rule specificity and rule generality has a direct influence on precision and recall.

Califf and Mooney (2004) describe different system designs of rule learning and rule induction: bottom-up, top-down, and their combination. Among them, the bottom-up rule induction is the most interesting for the minimally supervised or unsupervised rule learning systems. Thus, we will only focus on the bottom-up rule induction aspect, namely, deriving general rules from specific ones. There are two bottom-up rule induction methods: *compression* and *covering*. In the rule *compression* method, a system learns specific rules from each example. Given the specific rule set, it tries to derive the general rules iteratively. At each iteration, general rules are constructed which then replace the specific rules that are subsumed by them. One of the widely used subsumption checks is to evaluate whether the general rules can extract most positive examples supported by the specific rules and will not produce spurious examples. This subsumption check is called *empirical subsumption*, a notion introduced by the CHILLIN algorithm (Zelle and Mooney 1994). The iteration ends when no more new rules can be created. Systems that use *covering* begin with a set of positive examples. If a new rule is learned, all positive examples covered by this rule will be removed from the corpus for the future learning iteration. Rule learning ends when all positive examples have been covered. The generalization method in *compression* can be applied to the results delivered by *covering*.

Califf and Mooney (2004) take their RAPIER system (Robust Automatic Production of Information Extraction Rules) as an example of the rule compression method. RAPIER learns pattern-matching rules from annotated corpora. In

the following, we list the relevant parameters and the learning algorithm steps of RAPIER:

- **input**: corpus in a specific domain where the documents are aligned with extracted templates. The documents are preprocessed with PoS tagging. WordNet (Miller et al. 1998) is used for assigning semantic classes to the pattern units.
- **rule representation**: a tuple of three fields $\langle prefiller,\ filler,\ postfiller \rangle$
- **learning algorithm**
  - for each slot, most-specific patterns are created for each example
  - compress and generalize rules and specify rules
    1. new rules are created by selecting two existing rules and creating a generalization
    2. if the generalization produces spurious results, the prefiller and the postfiller of the original two rules will be used to specify the generalized rule
    3. if a general rule produces correct results, all rules subsumed by it will be removed
  - stop, if no new rule can be learned

The RAPIER rule generation method is specific for the RAPIER pattern representation. It takes information such as word length, ISA relation in WordNet and overlapping degree of rule elements into account. Figure 3.2 gives an example of the generation of two rule elements.

In Figure 3.2, Califf and Mooney (2004) depict that the words "man" and "woman" form two possible generalizations, namely, one is their disjunction and another is deletion of the word constraint. Furthermore, the tags "nn" (noun) and "nnp" (proper noun) have two possible generalizations too. Therefore, there are a total of four generalizations of the two elements. We will not discuss this method in detail here. Various generalization methods are investigated in the contexts of different rule learning systems depending on their rule representations. For example, the Snowball system applies a clustering method to compress the rules, while AutoSlog-TS relies on a set of generic linguistic patterns.

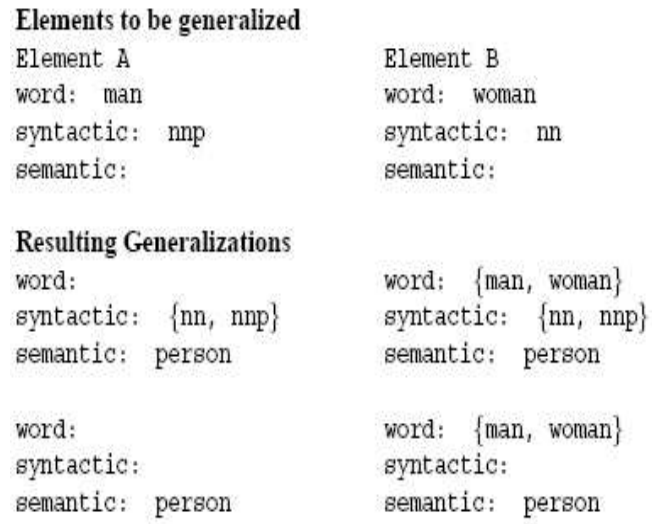

Figure 3.2: A RAPIER example of the generalization of two pattern elements

The central contribution of the RAPIER system is its generic rule induction strategy: compression, generalization, evaluation, specification and evaluation. The integration of an empirical subsumption check into a rule learning algorithm is very useful for controlling the quality of generalization.

3.4 Conclusion

We have presented different minimally supervised automatic pattern discovery systems: AutoSlog–TS, DIPRE, Snowball and ExDisco. DIPRE, Snowball and ExDisco start with some seed as initial domain knowledge and learn patterns in a bootstrapping manner. The duality and density principles provide the underlying parameters for controlling the quality of learned knowledge at each iteration. DIPRE and Snowball start with relation instances, while ExDisco relies on some domain-specific patterns. The pattern representations of DIPRE and the Snowball systems can be directly applied to relation extraction, because the relation instances in the seed set import the explicit semantic role information into patterns. But the pattern-based seed method of ExDisco only detects relevant linguistic patterns. The chain, linked chain and subtree models are faced with the same serious problem. Their patterns are incomplete for

relation extraction, because the role mapping information between linguistic arguments and semantic roles of the slot fillers is missing.

Stevenson and Greenwood (2006) compared various pattern representation models: SVO, chain, linked chain and subtree with respect to their productivity (enumeration of the rule number), their expressiveness and coverage. The SVO representation conducted by the ExDisco system has the poorest coverage and lowest rule number, while the subtree model achieves the highest coverage, but with the largest number of patterns. The huge number of rule patterns weakens the efficiency of the IE system enormously, and therefore yields an unacceptable method. The linked chain model has a very high coverage, but a relatively reasonable number of patterns.

Thus, given the large number of learned specific pattern rules, the rule induction and generalization strategy developed by Califf and Mooney (2004) is very useful. The RAPIER system is an example showing how to generalize rules and control the quality of the generalized rules via empirical subsumption check. A good tradeoff between specificity and generality of learned pattern rules is important for

- extraction performance: precision and recall,
- rule adaptability: general rules cope better with unseen data, and
- system efficiency: the larger the rule set, the higher the computational load.

We finish this chapter by stating the conclusions for our work. Our system design adopts the semantic-instance based seed idea of DIPRE and Snowball. Therefore, the learned pattern rules can straightforwardly be utilized as extraction rules. The duality principle is also the basic principle for our learning algorithm. We utilize the Ideal table proposed for Snowball for the evaluation of our own system. From the comparison of the different representation models and from the analysis of their respective shortcomings, we take on the challenge of developing a pattern representation method that meets the coverage and the expressivity requirements and that at the same time avoids the production of useless patterns as we saw in the subtree model. Our rule induction and generalization method is built on top of the bottom-up rule generalization idea.

Chapter 4

Preparatory Work

In this chapter, we present several pieces of our own work that directly contributed to the design or to the realization of the ***DARE*** framework. In Section 4.1, we describe a semantic model that defines IE as a pragmatic approach to the semantic understanding of texts. Domain modelling is conducted for the Nobel Prize award domain. Section 4.2 presents a method for exacting domain relevant terms from classified texts. A minimally supervised machine learning method is applied to the acquisition of lexico-syntactic patterns for detecting semantic relationships among terms. Section 4.3 explains a hybrid IE architecture that integrates both shallow and deep NLP for the template filling tasks. Two different pattern representations are developed to deal with the analysis delivered by the shallow and the deep NLP. In Section 4.4, we present a unification-based shallow NLP platform for recognition of named entities, terms and simple relations, which is employed as an analysis component for our relation extraction. Section 4.5 describes an IE application in question answering, utilizing the extracted semantic instances and modelled domain ontology.

4.1 A Semantic Model of an IE Task

If we consider IE as a pragmatic approach to semantic understanding, MUCs and ACE have identified the relevant components of the semantic model of IE tasks (Appelt 2003). Our definition is an extension of that suggested by Appelt (2003). An IE semantic model contains the following components:

- ***entities***: individuals in the world that are mentioned in a text
  - *simple entities*: singular objects
  - *collective entities*: sets of objects of the same type where the set is explicitly mentioned in the text
- ***relations***: properties that hold of tuples of entities
- ***complex relations***: relations that hold among entities and relations
- ***attributes***: one place relations that are attributes or individual properties
- ***temporal points or intervals***: relations that can be timeless or bound to temporal points or intervals
- ***events***: a particular kind of simple or complex relation among entities involving a change in at least one relation

A timeless attribute may be, at least in practice, the gender information of a person, while a time-dependent attribute could, e.g., be a person's age. The father and child relation is a timeless two place relation, while the employer and employee relationship is time-dependent.

In practice, the semantic tasks for IE are

- ontology modelling of the application domain:
  - definition of the relevant concepts and their IS-A and PART-OF relations
  - definition of the relevant domain-specific relations among the concepts
  - definition of relevant events
- recognition of linguistic entities
- classification of linguistic entities to their semantic classes defined in the ontology
- identification of equivalence classes of linguistic entities
- identification of linguistic relations

- classification of linguistic relations to their semantic classes
- classification of the linguistic entities to their semantic roles in relations or events

Except for the ontology modelling, the tasks mentioned above correspond to the components in the generic IE architecture already presented in section 2.4. They are: named entity recognition, coreference resolution, linguistic parsing, scenario pattern matching, etc. Thus, here we focus on the ontology modelling task and give an example of modelling the Nobel Prize award domain. The modelling method is described in Frank et al. (2006) and Xu et al. (2006).

In general, the modelling of the ontology of an application domain can start with an existing general ontology as reference. In addition to the ACE ontology and other resources, two useful resources could be considered as candidates: on the one hand the knowledge-engineering-based top-level ontology SUMO (Pease et al. 2002) and its mid-level specification MILO (Niles and Terry 2004) and on the other hand the structured thesaurus WordNet (Miller et al. 1998). Since there is a mapping between the artificial concepts in SUMO and the word senses in WordNet (Niles and Pease 2003), we have decided to choose the SUMO ontology as the backbone and define sub-concepts by referring to the mapping between SUMO concepts and WordNet word senses.

The main concepts in the application domain are *prize*, *laureate*, *area (of Prize)*, as well as domain-independent general concepts/entities, such as *date time*, *monetary value*, *organization* and *person*. Table 4.1 lists some of the mappings between domain concepts and SUMO concepts: *laureate* corresponds to the SUMO concept *cognitiveAgent*, inheriting therefore its two subconcepts *human* and *organization*. Most subconcepts of the concept *area*, except for *peace*, are subconcepts of the general concept *fieldOfStudy*, e.g., Chemistry. Each concept is further specified by its attributes. E.g., person is assigned the attributes first name and surname. The concepts are organized via hierarchical relations.

Along the lines of the entity types, the relations and the events can be organized in a IS-A relation too (Xu et al. 2006). For example, a general event type in the *award-winning* domain is called *receive-award*. Its arguments are

type	nobel-prize-winning domain	SUMO
entity	prize	award,...
entity	laureate	cognitiveAgent
entity	person	human
entity	organization	group
entity	area	fieldOfStudy, ...
event	receive-nobel-prize	unilateralGetting
event	nominate-nobel-prize	declaring, deciding

Table 4.1: Some samples of mapping between the domain ontology and SUMO

(4.1)
$$\begin{bmatrix} \textbf{recipients} & \text{a list of laureates} \\ \textbf{award} & \text{award-type (medal, prize, title, ...)} \\ \textbf{reason} & \text{achievement (accomplishment, service, skills, ...)} \\ \textbf{location} & \text{place} \\ \textbf{time} & \text{date time} \end{bmatrix}$$

The *receive-prize* event is a subtype of *receive-award*. Therefore, its arguments can be more specific:

(4.2)
$$\begin{bmatrix} \textbf{recipients} & \text{a list of laureates} \\ \textbf{award} & \text{prize name} \\ \textbf{reason} & \text{achievement (accomplishment, service, skills, ...)} \\ \textbf{area} & \text{area} \\ \textbf{location} & \text{place} \\ \textbf{prize amount} & \text{currency or percentage} \\ \textbf{time} & \text{date time} \end{bmatrix}$$

receive-Nobel-Prize is a subtype of *receive-prize*. Its arguments are then

(4.3)

recipients	a list of laureates
award	prize name (Nobel Prize)
reason	achievement (scientific contribution)
area	area (Nobel Prize discipline)
location	place
prize amount	currency or percentage
time	year

The ontology modelled for the *Nobel Prize award* domain is utilized in our thesis experiments presented in Chapter 6. This domain ontology is built on top of a general ontology and is thus scalable and better suited for the combination of domains. Due to the linking between SUMO and WordNet, the ontology has a direct access to WordNet via SUMO/MILO and thus supports application of lexical inference and the rule induction and generalization task.

4.2 Discovery of Domain Relevant Terms and Their Relations

Xu et al. (2002) have developed solutions to several IE problems: extraction of domain relevant terms, learning multi-word terms and collocations for free-word order languages, and minimally supervised learning of lexico-syntactic patterns for term relations. Because it falls outside the scope of this thesis, we will not present the multi-word term learning and the collocation method here.

4.2.1 Discovery of Domain Relevant Terms

The relevant term discovery method takes classified documents as input and applies a specific TFIDF measure (Salton 1991), called KFIDF, which is suitable when working with classified documents. The KFIDF is defined as follows:

$$KFIDF(w,c) = docs(w,c) \times LOG(\frac{n \times |c|}{c(w)} + 1) \tag{4.4}$$

docs(w,c)=number of documents in the class c containing the word w

n= smoothing factor

$c(w)$ = the number of classes in which the word occurs

According to this formula, the KFIDF measure for a word grows logarithmically inversely proportional to the number of categories it occurs in. In other words, a term is regarded as relevant if it occurs more frequently than other words in a certain class, but occasionally also elsewhere.

In our approach, only adjectives, nouns and verbs are considered as potential term candidates. We applied this measure to German newspaper texts taken from DPA (Deutsche Presse Agentur) in three domains: management succession, stock market and crime-drug domain.

An interesting phenomenon was observed. The distribution of the relevant terms concerning the part-of-speech (henceforth PoS) information is domain dependent. In some domains, the most relevant terms are nouns; for example, the drug crime domain and the stock market domain, while in some domains like management succession, the relevant terms are verbs.

Table 4.2 and 4.3 list the top ten noun terms extracted from the drug domain and the stock market document collection separately. In contrast to the above two domains, the management succession was determined mainly by the verbs which indicate the change of employment in company managements. In Table 4.4, some of the top terms are presented.

Term	**Score**
Haschisch (engl. hashish)	79.13055
Droge (engl. drug)	55.192017
Marihuana (engl. marihuana)	55.151592
Rauschgift (engl. drug)	53.61485
Kilogramm (engl. kilogram)	52.038185
Marktwert (engl. market price)	51.142445
Heroin (engl. heroin)	48.095898
Kokain (engl. cocaine)	44.153614
Schwarzmarktwert (engl. black-market price)	40.913956
Konsument (engl. consumer)	32.390213
Ecstasy-Tabletten (engl. ecstasy pills)	28.774744

Table 4.2: Top noun terms in the drug domain

We applied this term extraction and scoring method to the learning of relevant terms for our thesis experiment domains, i.e., Nobel Prize domain and man-

Term	Score
Aktienbörse (engl. stock exchange)	237.05634
Veränderung (engl. change)	143.48146
Gewinner (engl. winner)	142.09517
Verlierer (engl. loser)	142.09517
Hochtief (engl. up and down)	88.72284
Tief (engl. deep)	88.72284
Kugelfischer (engl. blowfish)	80.405075
Carbon (engl. carbon)	70.70101
Aktie (engl. stocks)	53.796547
Kurs (engl. stock price)	49.768997

Table 4.3: Top noun terms in the stock market domain

Term	Score
berufen (engl. appoint to)	38.45143
wählen (engl. choose)	35.155594
übernehmen (engl. accept)	32.95837
bestellen (engl. nominate)	28.56392
verlassen (engl. leave)	20.873634
wechseln (engl. change)	19.77502
ausscheiden (engl. resign)	17.577797
nachfolgen (engl. succeed)	15.380572
zurücktreten (engl. resign)	12.084735
antreten (engl. assume office)	8.788898

Table 4.4: Top verb terms in the management succession domain

agement succession. Table 4.5 shows the top terms in the Nobel Prize domain.

The English management succession corpus delivers results similar to the German corpus. Verbs dominate the proportion of the relevant terms. The relevant verbs include such words as *succeed*, *name*, *resign*, *acquire*, *retain*, *change*, *hire*, *step down*, *retire*, etc.

This method performs very well when the domains are clearly distinguished from each other. At least three domains are needed for achieving reliable results. In the reported experiments, we evaluated the top 100 terms for each domain. The average precision is above 80% for each relevant word class.

Term	Score	Pos
award	92.50355825813334	verb
win	73.12028947981658	verb
honour	30.76114408270707	verb
recognise	20.873633484694082	verb
nominate	15.970153924355433	verb
present	14.892097960034114	verb
share	13.298673069241266	verb
celebrate	11.127610028463279	verb
praise	10.067524107617551	verb
receive	9.215748569070595	verb
accept	8.615658321849084	verb
Nobel prize	128.53763777416881	noun
prize	124.14318861949639	noun
Nobel	107.66400428947473	noun
award	94.82594724025762	noun
winner	74.26605497979465	noun
peace	74.26605497979465	noun
Nobel Peace Prize	69.31891853645725	noun
honour	34.0569809487114	noun

Table 4.5: Top verb and noun terms in the Nobel Prize domain

4.2.2 Learning Patterns for Term Relation Extraction

Following the system design of DIPRE and Snowball (Brin (1998) and Agichtein and Gravano (2000)), we proposed an automatic method for learning lexico-syntactic patterns indicating paradigmatic relations (Hearst (1992) and Finkelstein-Landau and Morin (1999)). We employ the existing semantic relations provided by GermaNet (Hamp and Feldweg 1997) as initial knowledge and assign synonymy, hyponymy and meronymy relations among the terms in the corpus. GermaNet is the German development of WordNet (Miller et al. 1998). The terms are acquired by the term extraction component described in the section above. Then, we learn the lexico-syntactic patterns containing these semantic relations. Subsequently, we use the algorithm presented in Finkelstein-Landau and Morin (1999) for clustering similar patterns. The patterns are applied to extract new relation instances. In this approach, we classify the learned patterns into two groups: domain independent patterns and domain specific patterns. Domain specific patterns define reliable domain specific relations, for example (4.5).

(4.5) Drogen wie LIST_of_NPs (engl. *drugs like/such as LIST_of_NPs*)
Drogen z.B. LIST_of_NPs (engl. *drugs, e.g., LIST_of_NPs*)

The above patterns indicate that each single noun phrase (NP) in the above noun phrase lists (LIST_of_NPs) is a hyponym of Drogen (engl. *drug*), e.g., *Drogen wie Cannabis-Produkten*, where the *Cannabis-Produkt* is a hyponym of *Drogen*. The general lexico-syntactic patterns are as follows:

(4.6) NP, NP, ..., NP, NP und andere N (engl. *NP, NP, ..., NP, NP and other N*)
NP bzw. NP (engl. *NP and NP respectively*)
NP z.B. LIST_of_NPs (engl. *NP, e.g., LIST_of_NPs*)
NP wie LIST_of_NPs (engl. *NP like/such as LIST_of_NPs*)

Like the bootstrapping process in DIPRE and Snowball, this system applies the learned patterns to the corpus text again and extracts new related terms, which have potential hyponymy relations among them. For example (4.7), the three NPs in the LIST_of_NPs are hyponyms of the term *smuggling countries*. These hyponymy relations are very domain specific.

(4.7) Schmuggelländer wie [*Niederlande*, *Türkei* und *Ungarn*]$_{LIST\_of\_NPs}$
(engl. *smuggling countries such as Netherland, Turkey and Hungary*)

In many cases, we observed that large numbers of term groups do not have strict hyponym or synonym relations among themselves, for example,

(4.8) Kokain sowie Haschisch, LSD und Syntheseprodukt
(engl.*cocaine as well as hashish, LSD and synthetic product*)

Most of them are near synonyms (Inkpen 2001). Near synonyms are words that are almost synonyms, playing the same semantic role in a domain. They usually share a super concept. In order to identify their common super concept, we use GermaNet to search for their shared hypernyms. Afterwards, we assign the found hypernyms to the rest of the terms which are not encoded in the GermaNet. The advantage of this method is that we can assign the new terms

into the domain hierarchy and at the same time we disambiguate the senses of the terms in this domain.

In example (4.8), *Kokain* (engl. *cocain*) and *Haschisch* (engl. *hashish*) share the same super concept *Droge* (engl. *drug*) in GermaNet, therefore, we assign *Droge* (engl. *drug*) as the supper concept of *LSD* and *Syntheseprodukt* (engl. *synthesis product*). Many real-word applications, in particular, IE, typically require relatedness rather then just similarity. In the following example, the related terms are near synonyms in the criminal drug domain:

(4.9) a. Polizei, Zoll, Landeskriminalamt
(engl. *police*, *custom*, *state criminal investigation department*)
b. Schlaflosigkeit, Halluzinationen, Verfolgungswahn
(engl. *agrypnia*, *hallucination*, *persecution mania*)
c. Polizei, Drogenhilfe, Sozialarbeiter
(engl. *police*, *drug assistance*, *social worker*)

These clusters of terms correspond to special semantic concepts in the drug domain, (4.9a) to the concept *governmental institutions against drug traffic*, (4.9b) to the concept *side effects of drug consumption* and (4.9c) to the concept *aid organizations for drug addicts*.

In this approach, we adopt the system design of DIPRE and Snowball to extract ontological relations among relevant terms. This term relation extraction system utilizes a traditional pattern representation model, namely, the lexico-syntactic pattern. Although this representation allows the extraction of more than two arguments, it is still very surface-oriented and only suitable for extracting relations expressed in local or simple linguistic structures, such as noun phrase coordinations.

4.3 Hybrid NLP for Pattern Representation

In Xu and Krieger (2003), we describe an approach to IE by developing strategies for combining techniques from shallow and deep NLP. We propose a hybrid pattern representation strategy, which employs shallow partial syntactic analysis for extracting local domain-specific relations and uses predicate-argument

structures delivered by deep full-sentence analysis for extracting relations triggered by verbs. Heuristics are developed for triggering deep NLP on demand. The initial evaluation shows that the integration of deep analysis improves the performance of the scenario template generation task.

In current IE research, performance and domain adaptability are two essential issues. Regular expression based grammars (shallow grammars) embedded in IE systems, which employ finite-state techniques (Hobbs et al. 1997), subsumed under the term shallow NLP, often mix general linguistic information with domain-specific interpretation and are therefore not always portable. In addition, due to the inherent complexity of natural language, the same semantic relations can be expressed in different syntactic forms: in particular, via linguistic constructions, such as long distance dependencies, passive, control/raising. Such constructions are very hard to capture by pattern-based grammars. In contrast to shallow NLP, "traditional" full sentence analysis, called deep NLP, can, in principle, detect relationships expressed as complex constructions. Furthermore, most deep NLP systems are based on linguistically-motivated grammars, covering a huge set of linguistic phenomena. Such grammars should be more easily adapted to new domains and applications than the shallow grammars (Uszkoreit 2002). However, the scepticism of using deep NLP in real-life applications results from the lack of efficiency and robustness, and also from the huge number of ambiguous readings.

In the literature, there are several approaches to combing shallow NLP and deep NLP. In the large project Verbmobil (Wahlster 2000), the deep parser runs in parallel to the shallow and statistical parsing components, embedded in a concurrent system architecture. Tsujii (2000) briefly describes an experiment of applying the combination of shallow NLP and deep NLP to IE in the genome science domain. Riezler et al. (2001) present a stochastic system for parsing UPenn's Wall Street Journal (WSJ) treebank. The system combines full and partial parsing techniques by extending the full grammar with a grammar for fragment recognition.

Our system WHIES (WHiteboard Information Extraction System) is an attempt to combine the best of shallow and deep NLP and to keep the template-filling task independent of the general linguistic analysis. This system is built on top of an integrated system called WHAM (WHiteboard Annotation Machine), which provides access to both shallow and deep analysis results (Crysmann et al.

2002). WHIES takes partial syntactic analyses given by shallow NLP as the primary analysis and integrates deep results only on demand. Its hybrid template-filling strategy uses two kinds of template-filling rules: lexico-syntactic patterns and unification-based predicate argument structures. The pattern-based rules are applied to shallow NLP results in order to guarantee efficient and robust recognition of domain-relevant local relations. The unification-based rules are applied to predicate-argument structures, which result from full-sentence parsing done by the deep HPSG parser. Given typed feature structures as our basic data structure for template representation, the merging of partially filled templates is based on the unification operation. Template merging is handled as a two-step constraint resolution process at sentence and discourse level.

4.3.1 WHiteboard Annotation Machine (WHAM)

WHAM implements a hybrid system architecture for integrating shallow and deep NLP. WHAM provides access to linguistic analysis at different levels: tokens, morphological information, named entities, phrase chunks, sentence boundaries, and HPSG analysis results. The basic strategy in WHAM can be simply stated as "shallow-guided" and "shallow-supported" deep parsing. The integration takes place at various levels: lexicon, named entities, phrase level, and topological structure. A German text is at first analysed by SPPC, a rule-based shallow processing system for German texts, performing tokenization, morphological analysis, POS filtering, named entity recognition, phrase recognition, and clause boundary recognition (Piskorski and Neumann 2000). WHAM passes the shallow analyses for each sentence to a deep analyser, an efficient HPSG parser (Callmeier 2000) applied to the German grammar. The semantic analysis of the deep parser uses a kind of underspecified semantic representation, called MRS (Minimal Recursion Semantics) (Copestake et al. 2005).

4.3.2 Integration of Deep NLP on Demand

Shallow IE methods have been proven to be sufficient to deal with extraction of relationships among chunks, expressed relatively locally and explicitly (Grishman 1997). Normally, the interpretation of a sequence of chunks by shallow NLP is unambiguous and domain-specific, e.g., the relationships between a

noun phrase (NP) and its adjacent prepositional phrase (PP modifier) or its adjacent NP (appositive modifier). For deep NLP, the decision of the attachment of modifiers is very difficult, and thus, their analysis is often ambiguous. Nevertheless, deep grammars are more suitable for expressing precise relationships between verbs and their arguments in complex linguistic constructions, involving, e.g., passive, free word order, long-distance dependencies and control/raising. For example, sentence (4.10) contains a passive and a control construction. The relationship between the person name *Hans Becker* and the division name *Presseabteilung* (engl. *press division*) cannot be formulated easily by regular expressions. In particular, the relatively free word order of German allows reversing the order of the two names, while keeping the same meaning; see (4.11).

(4.10) *Hans Becker* wurde aufgrund des Rücktritts von *Peter Müller* gebeten, die Presseabteilung zu übernehmen.
(engl. *Hans Becker was due to the resignation of Peter Müller asked, to take over the press division.*)

(4.11) Aufgrund des Rücktritts von *Peter Müller* wurde *Hans Becker* gebeten, die Presseabteilung zu übernehmen.
(engl. *Due to the resignation of Peter Müller Hans Becker was asked, to take over the press division.*)

In comparison to most shallow approaches, our deep NLP system can recognize the embedded relationships in (4.10) and (4.11) straightforwardly, normalizing them into a predicate-argument structure. Although some of the shallow systems also perform full sentence analysis, most of them (like SPPC) provide only partial analysis and cannot capture these kinds of embedded relationships without any additional efforts.

Given the pros and cons of shallow and deep analysis, we decide to use shallow analysis as our primary linguistic resources for recognizing local relationships and have developed heuristics, which are used to trigger deep NLP only on demand.

As explained in the last section 4.2, a method (Xu et al. 2002) has been developed to recognize domain-relevant terms and their relations. Each term is

assigned a relevance weight. An interesting observation is that the distribution of relevant terms in a specific domain is related to the PoS information. For example, in the stock market and the drug crime domain, most relevant terms are nouns, while verbs play an important role in the management succession domain. This observation is a good indicator for deciding whether and when deep NLP should be integrated into IE for a new domain. If the domain-relevant terms are mostly verbs, we suggest integrating deep NLP for obtaining predicate-argument structures, since relationships triggered by the verbs can be expressed in various syntactic forms and therefore cannot easily be covered by a small set of pattern-based rules. For example, sentence (4.12) and (4.13) express the same meaning, but with different word order, as (4.10) and (4.11).

(4.12) Generaldirektor Eugen Krammer (59), ..., wird per 31. Mai 1997 aus seinen Funktionen ausscheiden.
(engl. *General manager Eugen Krammer (59), ..., will resign from his office on May 31. 1997*)

(4.13) Aus seinen Funktionen wird Generaldirektor Eugen Krammar (59)...., per 31. Mai 1997 ausscheiden.
(engl. *General manager Eugen Krammer (59), ..., will resign from his office on May 31. 1997*)

Both of them are about resignation of the person *Eugen Krammer*. The domain-relevant verb predicate *ausscheiden* (engl. *resign*) triggers the resignation relation, taking *Eugen Krammer* as argument. In this case, deep NLP can detect the predicate-argument structures in (4.12) and (4.13). Although (4.12) and (4.13) have different surface constructions, only a single rule has to be defined, which maps the argument of the predicate *ausscheiden* to its domain role.

In comparison to verbs, nouns (including nominalization of verbs) and adjectives are good indicators for pattern-based rules, which are suitable for dealing with local relationships expressed by complex noun phrases, containing PP-attachment and appositions. (4.14) and (4.15) give examples of adjectives and nouns as trigger words in the management succession domain.

(4.14) Der bisherige Vorstandsvorsitzende des Auto-Zulieferers Kolbenschmidt, Heinrich Binder, ...

(engl. *The previous president of car supplier Kolbenschmidt, Heinrich Binder ...*)

(4.15) Nachfolger vom Amtsinhaber Hans Günter Merk
(engl. *Successor of the office holder Hans Günter Merk*)

Thus, we take relevant verbs as clues for deciding when to trigger deep NLP during online processing: if a sentence contains relevant verb terms in addition to relevant nouns and adjectives, it will also be passed to deep NLP; otherwise, shallow NLP will be sufficient.

4.3.3 A Hybrid Rule Representation

The linguistic annotations provided by WHAM are domain independent. Our hybrid strategy allows for two kinds of template-filling rules, which map general linguistic analysis to domain-specific interpretations:

- lexico-syntactic pattern rules (*P-rule*)
- unification-based predicate argument structure rules (*U-rule*)

Here we use the management succession domain for presenting our ideas. P-rules are applied to shallow results, in particular to tokens, lexical items, named entities and phrases, using relevant adjectives and nouns as trigger terms. A P-rule consists of two parts: the left-hand side is a regular expression over typed feature structures, whereas the right-hand side is a typed feature structure, corresponding to a partially-filled scenario template, e.g.,

(4.16) Rücktritt von [1]Person → $\left[\text{PersonOut} \quad \boxed{1}\right]$

(4.16) matches an expression which contains two tokens, *Rücktritt* (engl. *retirement*) and *von* (engl. *of*), followed by a person name, and fills the slot PersonOut. *Rücktritt* is the trigger word. Applying (4.16) to the shallow analysis of sentence (4.10) and (4.11), the *PersonOut* slot of the template is then filled with the name *Peter Müller*. The *SProUT* system described in (Drożdżyński

et al. 2004) supports the definition of P-rules. We will explain *SProUT* in the next section. A *U-rule* makes use of the predicate-argument structures embedded in MRSs, provided by the deep HPSG parser. Hence, a U-rule might look like the following:

$$(4.17)\quad \begin{bmatrix} \text{Predicate} & \text{übernehm}(\textit{take over}) \\ \text{Agent} & \boxed{1} \\ \text{Theme} & \boxed{2} \end{bmatrix} \rightarrow \begin{bmatrix} \text{PersonIn} & \boxed{1} \\ \text{Division} & \boxed{2} \end{bmatrix}$$

Applying (4.17) to the deep analysis of (4.10) or (4.11), the *PersonIn* slot is filled with *Hans Becker* and the *Division* slot with *Presseabteilung*. In fact, our hybrid template-filling strategy can also be directly applied to a relatively deep shallow NLP system, which can provide predicate-argument structures in addition to fragments.

The initial evaluation shows that information extracted by P-rules and U-rules is complementary to each other. Their combination improves the expressiveness of the template filling rules in general. However, linguistic structures represented by P-rules are often arguments of the linguistic structures dominated by verbs. There is no mechanism developed in this approach to define or represent the linguistic relations between P-rules and U-rules. Therefore, this approach often delivers parallel partially filled templates within one sentence, or even within one clause, although these slot fillers can be directly exacted into one template, if their linguistic relationships are not ignored. Thus, an extra template merging component is needed to combine the partially filled templates at the sentence level and at the discourse level. The merging criterion is based on simple heuristics, namely, overlapping or distance. A further improvement is to take the linguistic and semantic relationships among the P-rules and U-rules into account to achieve more precise merging of template arguments.

4.4 *SProUT*

SProUT (Shallow Processing with Unification and Typed Feature Structures) (Drożdżyński et al. 2004) is a platform for development of multilingual shallow text processing and IE systems. The *SProUT* platform can be utilized to develop the generic IE architecture described in Section 2.4. It provides an

integrated grammar development and testing environment. The reusable core components of *SProUT* are a finite-state machine toolkit, a regular compiler, a finite-state machine interpreter, a type feature structure package, and a set of linguistic processing resources. The advantages of the *SProUT* system are that

- it allows a flexible integration of different processing modules in a cascaded system pipeline, such as tokenization, morphological analysis, named entity recognition and phrase recognition;
- it combines finite-state devices with unification-based grammars to achieve efficiency and expressiveness.

The finite-state devices are successfully applied to many real-world applications, in particular, in the IE applications. Systems like FASTUS (Hobbs et al. 1997), SMES (Neumann et al. 1997) and GATE (Cunningham 2002) are built on top of the finite-state technologies. In comparison to them, *SProUT* integrates the unification-based grammars to enable a better description of linguistic and domain-relevant phenomena and their relations. In our experiments, we employ *SProUT* for recognition of domain relevant entities or terms, and semantic relations among them.

The *SProUT* grammar formalism is called XTDL. It combines two well-known frameworks: regular expressions and typed feature structures. XTDL is built on top of TDL, a definition language for typed feature structures used as a descriptive device in several grammar systems (LKB (Copestake), PAGE (Uszkoreit et al. 1994), PET (Callmeier 2000)). The grammar elements of XTDL are organized in a type hierarchy where multiple inheritances are allowed. In Figure 4.1, we depict the definitions of some general linguistic types and their hierarchical relations: *sign* as a top linguistic type for all linguistic units, *morph* as a morphological analysis unit, *ne_type* as a named entity type, *ne_prize* as a type standing for all prize entities, *prize_area* representing the prize areas, and *t_relation* for term relations. The linguistic types such as *token*, *morph*, *ne_type* are subtypes of *sign*.

A grammar in *SProUT* consists of a set of XTDL rules, where the left-hand side is a regular expression over typed feature structures (TFSs), representing the recognition pattern, and the right-hand side a TFS, specifying how the output structure looks. A XTDL rule has in general the following format:

```
      sign := *avm* & [ SURFACE      string,
                        CSTART       string,
                        CEND         string ].
     morgh :=   sign  & [ POS        pos,
                        STEM         string,
                        INFEL        infl ].
   ne_type :=   sign  & [ NECSTART   string,
                        NECEND       string,
                        DESCRIPTOR   string ].
  ne_prize := ne_type & [ PRIZENAME  string ].
 prize_area := ne_type & [ CLASSIFY  string ].
 t_relation :=  sign  & [ ARG1       ne_type,
                        ARG2         ne_type,
                        ARG3         ne_type ].
```

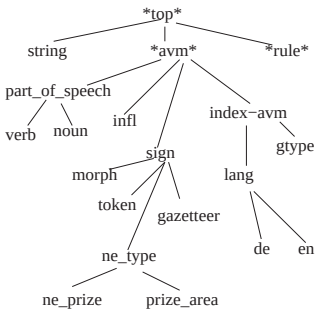

Figure 4.1: Examples of type hierarchy in *SProUT* and a type hierarchy in *SProUT*

(4.18)
```
rule_name :>  (regular expressions over TFSs) -> (TFS).
```

The following example extracts an event containing three arguments: the *prize name*, the *prize area* and the *event year*:

(4.19)
```
prize_area_time_relation :>
        (morph & [ SURFACE "the"] | morph & [SURFACE "a"]
        | morph & [ SURFACE "The"]   | morph & [SURFACE "A"])
        @seek(en_year) & #time
        gazetteer & [GTYPE gaz_prize, CONCEPT #id, CSTART #c1, CEND #c2]
        gazetteer & [GTYPE gaz_area_science,  CONCEPT #area,
                     CSTART #c3, CEND #c4]
        gazetteer & [GTYPE gaz_prize_word]?
    ->t_relation & [ARG1 ne_prize & [PRIZENAME #id, CSTART #c1,CEND #c2],
                   ARG2 prize_area & [CLASSIFY #area, CSTART #c3, CEND #c4],
                   ARG3 #time].
```

This rule can extract the event arguments from a local textual fragment, such as a noun phrase compound below:

(4.20) the *1999 Nobel Peace Prize*

The symbol `#` expresses the coreference relationships among the arguments. `gaz_prize` and `gaz_area_science` are elements in the gazetteer lists for prize names and scientific areas. *SProUT* allows users to add different gazetteer lists to the grammars. All gazetteer types are subtypes of the predefined *SProUT* type `gtype`. Entries in the gazetteer list look like the following:

(4.21)
```
Nobel | GTYPE:gaz_prize | CONCEPT:nobel | LANG:en
Pulitzer | GTYPE:gaz_prize | CONCEPT:pulitzer | LANG:en
```

The words *Nobel* and *Pulitzer* will be recognized as the `gaz_prize` type and are semantic concepts *nobel* and *pulitzer*. This *SProUT* gazetteer approach facilitates the definition of multilingual variants of same semantic concepts.

Mohamed ElBaradei, won the 2005 Nobel Peace Prize on Friday for his efforts to limit the spread of atomic weapons.

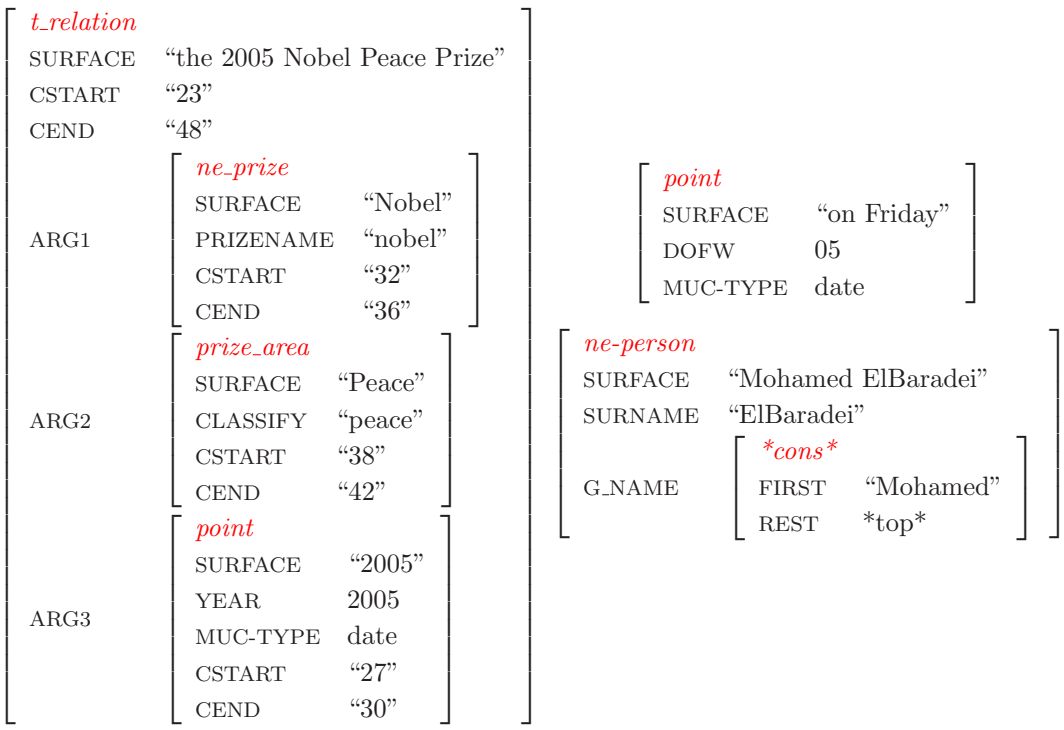

Figure 4.2: Examples of *SProUT* outputs

In Figure 4.2, we show an example of *SProUT* output. All *SProUT* examples presented above are provided by Li (2006).

In general, *SProUT* provides a powerful grammar formalism for formulating pattern action rules of an IE task. The combination of regular expressions and type feature structures turns out to be a convenient representation method of supporting the general shallow pattern matching and relation extraction task.

4.5 Querying Domain-Specific Structured Knowledge Resources

In this section, we present an application of the IE results, namely, question answering of structured knowledge resources (Frank et al. 2006). The events and relations extracted for the Nobel Prize winner domains are employed as one of the knowledge resources. The modelled ontology presented in Section 4.1 contributes to the inferencing task.

We present an implemented approach for domain restricted question answering from structured knowledge sources, based on robust semantic analysis in a hybrid NLP system architecture. We build on a lexical semantic conceptual structure for question interpretation, which is interfaced with domain-specific concepts and properties in a structured knowledge base. Question interpretation involves a limited number of domain-specific inferences. We extract so-called proto queries from the linguistic representation, which provide partial constraints for answer extraction from the underlying knowledge sources.

Example (4.22) shows the question interpretation and its proto query translation. The question interpretation makes use of the linguistic analysis delivered by the hybrid NLP system HOG (Callmeier et al. 2004) and the frame semantic information (Baker et al. 1998). It identifies the question focus (*q_focus*), expected answer type (*EAT_rel*), relevant entity objects (such as *person* and *area*) and semantic frames (e.g., *getting* and *award*). The question interpretation delivers input for the proto query translation. The proto query can be easily converted into a SQL query to access the data records in a relational database.

(4.22) 1. *In which areas did Marie Curie win a Nobel prize?*

2. Question interpretation

$$\begin{bmatrix} \text{REL} & q\_focus \\ \text{ARG0} & x10 \end{bmatrix} \begin{bmatrix} \text{REL} & EAT\_rel \\ \text{ARG0} & x10 \\ \text{SORT} & FieldofStudy \end{bmatrix}$$

$$\begin{bmatrix} \text{REL} & person \\ \text{ARG0} & x17 \\ \text{CARG} & Marie\ Curie \end{bmatrix} \begin{bmatrix} \text{GETTING} & e2 \\ \text{THEME} & x21 \\ \text{RECIPIENT} & x17 \end{bmatrix} \begin{bmatrix} \text{AWARD} & x21 \\ \text{LAUREATE} & x17 \\ \text{DOMAIN} & x10 \end{bmatrix}$$

$$\begin{bmatrix} \text{LAUREATE} & x17 \end{bmatrix}$$

3. Proto Query

```
<PROTO-QUERY id="1">
  <SELECT-COND qid="0" rel="award" attr="domain"
            sort="FieldofStudy">
  <WHERE-COND qid="0" rel="award" attr="laureate"
           netype="person" val="Marie Curie">
</PROTO-QUERY>
```

The instances of domain relations are stored in a relational database (MySQL). We store the Nobel prize winners in two separate tables: one for persons and one for organizations, because the two concepts (person and organization) are associated with different information. In the following examples, we call them "winner-person" and "winner-organization".

The first task of answer extraction is to take the proto queries provided by the question analysis as input and translate these into SQL queries. As explained, the proto queries identify

- the answer type concept, which corresponds to the value of the SQL "select"-command
- additional concepts and their values, which constrain the answer type value. These concepts will fill the SQL "where" conditions
- dependencies between elementary questions, if a question is complex and needs to be decomposed into several simple questions

For example, for a simple fact-based question such as "Who won the Nobel Prize in Chemistry in 2000?", the question analysis constructs a proto query (as stated below):

```
<PROTO-QUERY id="q13" type="sql">
  <SELECT-COND rel="award" attr="laureate"/>
  <WHERE-COND rel="award" attr="domain" val="Chemistry"/>
  <WHERE-COND rel="award" attr="time" val="2000"/>
</PROTO-QUERY>
```

Figure 4.3: Proto query for *Who won the Nobel Prize in Chemistry in 2000?*

The above query corresponds to a partially filled scenario template, where some slots are marked as queried objects ("?"):

$$\begin{bmatrix} \textsc{Laureate} & ? \\ \textsc{Area} & \textit{Chemistry} \\ \textsc{Year} & \textit{2000} \end{bmatrix}$$

The task of sql-query translation is to identify at first right tables where the information can be found and the right table fields which can match the values given in the proto-query.

We defined mapping rules between the FrameNet node and its argument and the database tables and their fields. The event specific fields are assigned with "yes". Here are some examples:

Rel	Attr	val–concept	DBTable	DBField	event–dependent
award	laureate	person	winner-person	name	yes
award	laureate	organization	winner-organization	name	yes
award	domain	prize-area	winner-person	area	no
award	domain	prize-area	winner-organization	area	no
award	time	date time	winner-person	year	yes
award	time	date time	winner-organization	year	yes

Table 4.6: Mapping table between FrameNet and knowledge resource

As we saw, in *SELECT-COND*, we have only the information of *rel* and *attr*. There is no direct table for *laureate*. In this case, we use our ontology information, namely, laureate corresponds to *cognitiveAgent* which has two subconcepts *human* and *group*. Their corresponding domain concepts are *person* and *organization*. We expand the values of *laureate* to *person* and *organization* and then find the potential tables. In the same way, we also search the tables for

the *WHERE-COND*. In this example, *SELECT-COND* and *WHERE-COND* share the same tables. Therefore, we generate the following two SQL-queries from the proto-query:

- **select** name **from** nobel-prize-winner-person
  textbfwhere year="2000" AND area="chemistry"
- **select** name **from** nobel-prize-winner-organization
  where year="2000" AND area="chemistry"

The final answer is obtained by merging the results from 1 and 2.

```
<PROTO-QUERY id="1">
  <SELECT-COND rel="award" attr="time" sort="Year"/>
  <WHERE-COND rel="award" attr="domain" netype="prize-area"
              val="Literature"/>
  <WHERE-COND rel="award" attr="laureate" netype="person"
              val="Nadine Gordimer"/>
</PROTO-QUERY>
```

Figure 4.4: Proto query for *In which year did Nadine Gordimer win the Nobel prize for Literature?*

In this proto query, both the *SELECT-COND* and the first *WHERE-COND* identify the two tables *winner-person* and *winner-organization*. In the second *WHERE-COND*, the linguistic analysis identifies the entity type of **laureate** as *person*. Therefore, we can use this information for table disambiguation and choose the table *winner-person* as our table.

The SQL query for this question is then:

select year **from** winner-person **where** area="Literature" AND
name="Nadine Gordimer"

In our approach, we handle queried entities independent of individual prize winning events differently from event dependent entities.

Let us compare the following two questions:

(4.23) 1. *In how many areas has France won a Nobel Prize?*

2. *How many Nobel Prize winners has France produced?*

In the first case, every area in which French persons or organizations have received a Nobel prize is counted once. For answering the second question, we could count every person once, even if the person has been awarded two prizes such as Marie Curie. However, we decided to make the cardinality of recipients event dependent, in line with counting tourists to Paris or customers of Harrod's. Thus the answer to the first question will be:

six areas: Chemistry, Physics, Peace, Literature, Medicine, Economics

Although all areas occur more than once, e.g. there are two French prizes for economics, we handle "area" as event independent. Our SQL-query will look like this:

select distinct area from TABLE where country="France"

The answer to the second question will be

"53 winners"

followed by the list of prize winners. Here Marie Curie would be counted twice. Thus the person in "award" relation is handled as event dependent.

Therefore, our SQL query is

select person from TABLE where country="France"

The QA method presented here is embedded in a hybrid QA architecture called QUETAL, developed by the QUETAL project[1]. The QUETAL architecture combines domain-specialized and open-domain QA techniques, accessing structured, semi-structured and unstructured data and knowledge. Due to lack of space, we will not explain other QUETAL functions in this work.

[1] `http://quetal.dfki.de`

An initial evaluation is conducted to assess the advantages of structured knowledge resources in comparison to a web-based open-domain textual question answering system AnswerBus (Zheng 2002). We compiled a set of 100 English questions about the Nobel prize domain, in part adapted from or inspired by the FAQ sections of Nobel prize web portals.

The question types in our test set range from factual and list questions to different types of cardinality and quantificational questions. The questions vary in terms of paraphrases (verbal and nominal paraphrases, interrogative, non-interrogatives or embedded questions, e.g.,

(4.24) *Give me a list of ...*
Could you tell me in which year ...,

and according to different types of constraints to be used in question interpretation and answer extraction, such as (relational) temporal constraints such as

in/before/since/after 1999,

and gender, prize areas, as well as countries, locations, and affiliations.

We evaluated the answer extraction module on the basis of the 58 correct proto queries that were selected by the voting procedure. For 74.1% of the proto queries the correct answer was returned; in 6.9% the answer was wrong; for 19%, finally, no answer was returned. Error analysis for the 4 incorrect answers yielded a single minor cause of error (wrong answer type identification). For missing answers we identified several causes that need to be adjusted: mismatches of concept-database mappings, wrong table selection and out of scope phenomena.

We collected the three highest-ranked answers returned by AnswerBus, and evaluated the returned answers. The coverage on our 100 question sample of AnswerBus is rather poor: it delivered a correct answer within the first three ranks for only 15% of the questions. Detailed analysis of the distribution of results over question types shows that AnswerBus fares moderately well for factual questions, but shows poor performance for other question types, such as

cardinality, quantificational, or embedded questions. Of the remaining question types, none could be answered.

IE enables the extraction of structured knowledge from unstructured textual data in an offline mode. Fleischman et al. (2003) have also presented an approach that integrates the offline extracted facts into an existing QA system. The improvement of QA performance and efficiency is impressive. Our experiment additionally confirms that the offline extracted structured knowledge is more suitable for providing precise and exact answers and in particular handling question types such as list, cardinality and quantification in addition to the factual question type.

4.6 Conclusion

In this chapter we present results from our own previous work that have contributed to the research for this thesis. Some of the methods and tangible results are integrated into our ***DARE*** system, while other studies helped us gain insights that turned out to be relevant for the core of the thesis.

The IE semantic model of an application domain provides a clear semantic structure among entities, relations and events. A semantic model is defined for the Nobel Prize award domain. Our domain ontology provides on the one hand the links between the domain relevant entities and the general SUMO concepts and on the other hand the access to the general lexical ontology WordNet.

The classification-based relevant term extraction discovers the relevant terms quite effectively. The extracted terms and their relevance are integrated into the ***DARE*** scoring method for pattern rules. The observation, that the distribution of word classes of relevant terms is domain dependent, confirms that the pattern representation should be expressive enough to cover all word classes that contain relevant relation trigger words.

The minimally supervised pattern learning method is applied to discover ontological relationships among terms. Since it learns ontological relation instances from the corpus automatically, it is useful for updating and enhancing the domain ontology. As a side product of this method, the acquired near synonyms assign relevant terms with their domain-specific interpretations. Above

all, building this system was a valuable exercise for conducting semantic seed based system design. However, the obtained lexico-syntactic patterns are still too surface-oriented and are not suitable for extracting complex semantic relations.

In the research context of the WHIES system development, we discussed the disadvantages of IE systems entirely dependent on shallow analysis. A solution is proposed that employs hybrid template filling rules: lexico-syntactic pattern rules and the predicate argument structure rules. Although the combination achieved a better coverage than a single pattern representation, this approach does not provide a mechanism for setting up the linguistic relationships between the two representations. Therefore, the extraction results are isolated from each other, even if they are linked with each other via their linguistic structures. Thus, the combination of the partially filled templates at claus or sentence level also have to be solved by the template merging component, which is originally assumed to operate at the discourse level.

The *SProUT* system is a shallow multilingual platform for IE system development. The combination of finite state devices and the typed feature structures enables a shallow system to be both efficient and expressive. The definition of XTDL rules for named entity or term recognition, even simple relationship recognition, is much more convenient than other shallow NLP platforms. *SProUT* is an important NLP tool in our ***DARE*** system, applied to named entity and term recognition.

Although QA is not the focus of our research, we show one application example of the learned relation instances, namely, question answering based on structured knowledge resources. As already proven by various QA systems, integration of structured knowledge resource leads to clear improvement of answer quality. In our experiment, we also demonstrate that the structured knowledge resources enable the system to deliver answers to question types such as cardinality, quantification, etc.

Chapter 5

*D*omain *A*daptive *R*elation *E*xtraction Based on Seeds: the *DARE* System

In this chapter, we describe our own approach, i.e., a minimally supervised machine learning framework for extracting relations of various complexity, called ***DARE***. The bootstrapping starts from a small set of n-ary relation instances as "seeds", in order to automatically learn pattern rules from parsed data, which can then extract new instances of the n-ary relation and its projections. We present a novel rule representation model which enables the composition of n-ary relation rules on top of the rules for projections of the relation. The compositional approach to rule construction is supported by a bottom-up pattern extraction method. The whole approach is implemented as the ***DARE*** system. We start with an overview of the problems and challenges in the current state of the art in section 5.1. We describe the algorithm for rule learning and relation extraction in section 5.2. Since *seed* plays an important role in this approach, a special section 5.3 is devoted to the seed idea. The rule representation model is explained in section 5.4. In section 5.5, we give a detailed description of the ***DARE*** system architecture and its components for relevant text snippet retrieval, pattern extraction, rule induction, rule application and ranking and filtering methods for validation of new rules and new extracted instances. In the conclusion section 5.10, we give a summary of the advantages of the ***DARE*** approach.

5.1 Motivation

As discussed in Chapter 3, current minimally supervised or unsupervised approaches to automatic pattern acquisition are still faced with the following problems:

- lack of linguistic expressiveness
- lack of semantic richness
- no systematic method for handling the linguistic interaction between relations and their projections
- no systematic method for handling relations of various complexities

In Yangarber (2001), the extraction pattern is limited to the subject-verb-object construction. Sudo et al. (2003) and Greenwood and Stevenson (2006) have improved the linguistic expressiveness of their pattern representation models, taking additional linguistic structures into account. Stevenson and Greenwood (2006) present a systematic investigation of the pattern representation models and point out that substructures of the linguistic representation and access to the embedded structures are important for obtaining a good coverage of pattern acquisition. However, all considered representation models (subject-verb-object, chain model, linked chain model and sub-tree model) are verb-centered. Relations embedded in non-verb constructions such as compound nouns cannot be discovered (see example (5.1)).

(5.1) the 2005 Nobel Peace Prize

(5.1) describes a ternary relation referring to three properties of a prize: year, area and prize name.

Sudo et al. (2003) attempts to cover as many verb-centered subtree structures as possible and has a severe computational problem in handling the large number of subtree patterns. The hybrid rule representations proposed by WHIES (Xu and Krieger 2003) attempt to cover as many relevant linguistic structures as possible. However, there is no mechanism developed to combine the two representations, which is important in dealing with relation extraction with various complexities.

We also observe that the automatically acquired patterns in (Riloff (1996), Yangarber (2001), Sudo et al. (2003), Greenwood and Stevenson (2006)) cannot be directly used as relation extraction rules because the relation-specific argument role information is missing. E.g., in the management succession domain that concerns the identification of job changing events, a person can either move into a job (called PersonIn) or leave a job (called PersonOut). (5.2) is a simplified example of patterns extracted by these systems:

(5.2) ⟨subject: person⟩ verb ⟨object: organization⟩

In (5.2), there is no further specification of whether the person entity in the subject position is PersonIn or PersonOut.

None of the approaches mentioned above considers the linguistic interaction between relations and their projections on k dimensional subspaces where $1 \leq k < n$, which is important for scalability and reusability of rules. Therefore, there is no systematic method for handling relations with various complexities.

In order to cope with the problems mentioned above, we work out the following solutions for the ***DARE*** framework:

- semantically oriented seed construction
- seed-driven bottom-up automatic pattern acquisition and rule composition strategy
- the compositional rule representation model
- exact assignment of semantic roles to the slot fillers in the extraction rules
- cascaded bottom-up rule induction: redundancy deletion, compression and generalization
- ranking method for new rule and new seed instance validation
- top-down rule matching for relation extraction
- fusion of partial relation projections

5.2 Algorithm

As mentioned above, our system learns rules from un-annotated free texts, taking some seed relations or events in the initialization. The learned extraction rules are then applied to the texts for detection of more relation and event instances. The newly discovered relations become new seeds for learning more rules. The learning and extraction processes interact with each other and are integrated in a bootstrapping framework. The whole algorithm works as follows:

1. **Input**:
   - A set of un-annotated free natural language texts
   - A trusted set of relation instances, initially chosen ad hoc by the users, as *seed*.
2. **Text/Passage retrieval**: Apply seeds to the documents and divide them into relevant and irrelevant documents. A document is relevant if its text fragments contain a minimal number of the relation arguments of a seed and the distance among individual arguments does not exceed the defined width of the textual window.
3. **Pattern extraction**: Annotate the relevant text fragments with named entities and linguistic structures and extract linguistic patterns which contain seed relation arguments as their linguistic arguments.
4. **Rule induction**: Induce relation extraction rules from the set of patterns using compression and generalization methods.
5. **Rule Ranking**: Rank the rules based on their domain relevance and the trustworthiness of their origin
6. **Relation extraction**: Apply induced rules to the corpus, in order to extract more relation instances.
7. **Ranking and validation**: Rank and validate the new relation instances.
8. **Stop** if no new rules and relation instances can be found, else repeat step 2 to step 6.

5.3 Seed

Many minimally supervised machine learning IE systems based on bootstrapping are initialized with so-called seeds. There are two general directions of seed construction:

- pattern based
- semantics (relation instance) based

The pattern oriented approaches take linguistic patterns as seeds. In ExDisco (Yangarber 2001), some example patterns of the management succession domain are chosen as the seed, e.g.,

(5.3) subject(company) verb("appoint") object(person)

for learning more relevant patterns which co-occur with the seed patterns in the documents and use new patterns as new seeds. However, new patterns generated by this class of methods are only relevant patterns for the training domain. They do not contain required information, namely,

- which kind of relation type they indicate and
- which semantic role the linguistic argument should be assigned to.

Thus, these patterns cannot be directly used as relation extraction rules. An additional obvious disadvantage of this pattern-oriented seed approach is that it is too closely bound to the linguistic representation of the seed. It is well known that semantic relations and events could be expressed via different levels of linguistic representations that do not restrict the realizations to one or more patterns such as *subject verb object* constructions. Furthermore, an event can be more complex than can be expressed by one single pattern. Moreover, most of these linguistic patterns only extract one or two arguments of a relation.

Thus, we favor a semantics-oriented notion of seed construction, using relation and event instances as our seeds, such as the DIPRE system (Brin 1998) and the Snowball system series (Agichtein et al. 2000) and (Agichtein and Gravano 2000). The advantages of this seed construction method are

- domain independence: it can be applied to all relation and event instances
- flexibility of the relation and event complexity: it allows n-ary relations
- processing independence: the seeds can lead to patterns in different processing modules, thus also supporting hybrid systems, voting approaches etc. and
- not limited to a sentence as an extraction unit.

The seed in our approach can fulfill or support the functions of

- detection of relevant sentences and passages which describe the seed relations. The relevant sentences can be used as potential event and relation extent
- detection of relevant linguistic expressions, which can be used as event and relation triggers
- detection of linguistic expressions, which fill a subset of the relation arguments
- detection of the interaction rules among patterns for relation projections, how they contribute to one complex relation

It is important for an unsupervised learning system to know the complexity and the structure of a seed in order to find good candidates for learning good extraction patterns and their interaction. Assume we want to build a database about paintings which provides information about dates of creation of paintings. Let us consider the following seed options:

(1) $< painter,\ creation\_year >$

(2) $< painter,\ painting >$

(3) $< painter,\ painting,\ creation\_year >$

(4) $< painter,\ painting,\ creation\_year,\ birth\_year >$

The seed suggestion in (1) is very underspecified and is not suitable for detecting patterns which indicate the creation date of a special painting, because

many relations and events can have a person name and a certain year as their arguments. Taking (1) as seed requires shifting the disambiguation to the later components. Although a painter and his painting can be involved in various relations and events such as "liking", "selling", that have nothing to do with the painting creation, the chance with (2) as a seed of finding the creation date seems larger than (1), because *painting* is more specific than *year*. (3) explicitly contains all three arguments. The most probable relation among them is the creation year of the painting by a painter. Therefore, (3) seems to be a trustworthy seed relation instance. However, it is not true that the more arguments a relation instance contains the better the instance is as a seed candidate. For example, if we take (4) as a seed with the addition of birth year information, relevant texts without birth year information will get lost. Thus, a tradeoff must be found with respect to the complexity.

In our approach, we choose the smallest number of arguments which together most probably express the relation. Furthermore, we take only relation instances into account which represent the relation type unambiguously, because there are some domains where the same argument tuple can present different relations. For example in the management succession domain, one person can take over a job in a company (called *PersonIn* relation) and resign from the same job in the same year (called *PersonOut* relation). If we take this person name, the job name, the company name and the year time as our seed, we will find a set of patterns for both relations. Thus, it is important to find the right combination of arguments which leads to learning unambiguous patterns.

5.4 Compositional Rule Representation Model

We propose a compositional rule representation model which supports bottom-up rule composition. A rule for an n-ary relation can be composed of rules for its projections, namely, rules that extract a subset of the n arguments. In comparison to previous pattern representations mentioned by Stevenson and Greenwood (2006), the ***DARE*** model is much more expressive for the representation of rules of various complexity. Furthermore, it defines explicitly the semantic roles of linguistic arguments for the target relation. Given the linguistic annotation, the rule specifies the precise linguistic relationship among the relation arguments. As a side effect, the rules for the projections may be

reusable for other relation tasks.

DARE rules are not restricted to a particular linguistic representation and are adaptable to various and even hybrid NLP tools on demand. A simple regular expression rule that applies to a Nobel Prize domain example in (5.4) is shown in (5.5). This rule recognizes the triple relation $\langle Prize, Area, Year \rangle$ in a noun phrase compound. The rule in (5.6) is a much more complex rule. It utilizes a *subject verb object* function triggered by the verb *win* and calls further rules for recognition of relation arguments that are embedded in the subject and the object: `recipient_1` and `prize_area_year_1`. `recipient_1` can be rules for recognizing person names. `prize_area_year_1` is described in (5.5). Assuming that all named entities such as person names, years, area names and prize names are recognized beforehand, the application of (5.6) to (5.4), yields the result in (5.7).

(5.4) *Mohamed ElBaradei*, won the *2005 Nobel Peace* Prize on Friday for his efforts to limit the spread of atomic weapons.

(5.5) `Rule name::` `prize_area_year_1`

`Rule body::`

$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{noun} \\ \text{lex-form} & \text{“prize”} \end{bmatrix} \\ \text{daughters} & \langle \begin{bmatrix} \text{lex-mod} & \begin{bmatrix} \text{head} & \boxed{3}\ \text{Year} \end{bmatrix} \end{bmatrix}, \\ & \begin{bmatrix} \text{lex-mod} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Prize} \end{bmatrix} \end{bmatrix}, \\ & \begin{bmatrix} \text{lex-mod} & \begin{bmatrix} \text{head} & \boxed{2}\ \text{Area} \end{bmatrix} \end{bmatrix} \rangle \end{bmatrix}$$

`Output::` $\langle \boxed{1}Prize, \boxed{2}Area, \boxed{3}Year \rangle$

(5.6) `Rule name::` `recipient_prize_area_year_1`

`Rule body::`

$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{verb} \\ \text{mode} & \text{active} \\ \text{lex-form} & \text{“win”} \end{bmatrix} \\ \text{daughters} & \left\langle \begin{bmatrix} \text{subject} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Person} \\ \text{rule} & \text{recipient\_1::}\ \langle \boxed{1} Person \rangle \end{bmatrix} \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{object} & \begin{bmatrix} \text{head} & \begin{bmatrix} \text{lex-form} & \text{“prize”} \end{bmatrix} \\ \text{rule} & \text{prize\_area\_year\_1::}\ \langle \boxed{2} Prize, \boxed{3} Area, \boxed{4} Year \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

`Output::` $\langle \boxed{1} Recipient, \boxed{2} Prize, \boxed{3} Area, \boxed{4} Year \rangle$

(5.7) $$\begin{bmatrix} \textbf{recipient} & \boxed{1}\ \text{Mohamed ElBaradei} \\ \textbf{prize} & \boxed{2}\ \text{Nobel} \\ \textbf{area} & \boxed{3}\ \text{Peace} \\ \textbf{year} & \boxed{4}\ 2005 \end{bmatrix}$$

A ***DARE*** rule is allowed to call further ***DARE*** rules which extract a subset of the arguments it has to exact. The syntax of a ***DARE*** rule is defined as follows:

Definition 1 (Syntax of a *DARE* rule)
A ***DARE*** rule has three components:

1. `rule name`: r_i;
2. `output`: a set A containing n arguments of the n-ary relation, labelled with their argument roles;
3. `rule body`: an AVM containing:
   - `head`: the linguistic annotation of the top node of the linguistic structure;
   - `daughters`: its value is a list of specific linguistic structures (e.g., `subject`, `object`, `head`, `mod`), derived from the linguistic analysis, e.g., dependency structures and the named entity information;
   - `rule`: its value is a ***DARE*** rule which extracts a subset of arguments of A.

The rule (5.6) is a typical ***DARE*** rule. Its subject and object trigger corresponding ***DARE*** rules which extract a subset of its output relation arguments. The constraints are formulated in the rule body.

As discussed in the description of the WHIES system (see Section 4.3), the regular expression rules (shallow NLP) are more suitable for local structures such as noun phrase compounds, appositions, noun phrases with PP attachments, while the grammatical functions and dependency structures (deep NLP) are useful in identifying relational arguments which are not adjacent to each other.

The advantages of this rule representation strategy are that

- it supports the bottom-up rule composition;
- it is expressive enough for the representation of rules of various complexities;
- it reflects the precise linguistic relationship among the relation arguments and reduces the template merging task in the later phase;
- the rules for the subset of arguments may be reused for other relation extraction tasks.

The rule representation models for automatic or unsupervised pattern rule extraction discussed by Stevenson and Greenwood (2006) do not account for these points.

5.5 System Architecture

The ***DARE*** architecture has been inspired by several existing seed-oriented unsupervised machine learning systems, in particular by Snowball (Agichtein and Gravano 2000) and ExDisco (Yangarber 2001). ***DARE*** contains four major components: *linguistic annotation*, *classifier*, *rule learning* and *relation extraction*. The first component is only applied once, while the last three components are integrated in a bootstrapping loop. At each iteration, rules will be learned based on the seed and then new relation instances will be extracted by applying the learned rules. The new relation instances are then used as seeds for the

next iteration of the learning cycle. The cycle terminates when no new relation instances can be acquired.

The ***DARE*** system architecture is depicted in Figure 5.1.

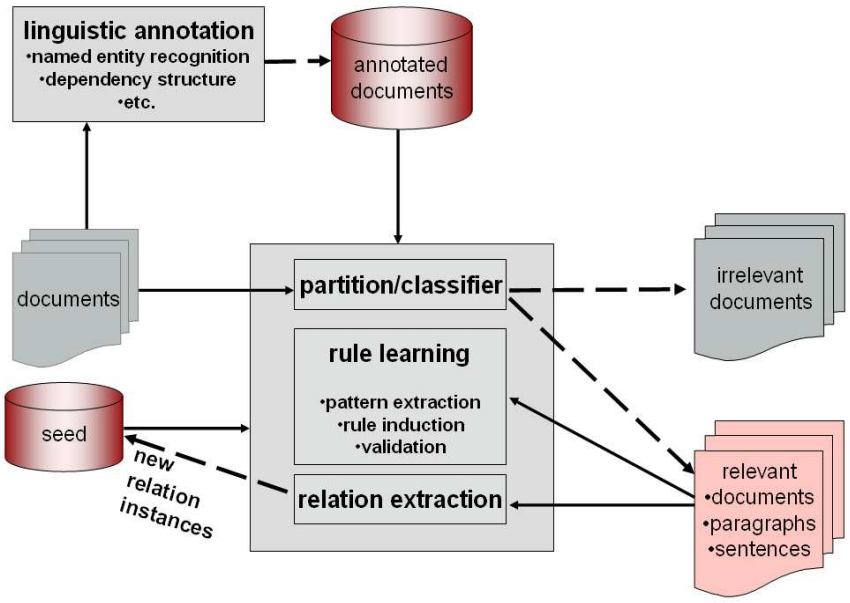

Figure 5.1: ***DARE*** Architecture

The **linguistic annotation** is responsible for enriching the natural language texts with linguistic information such as named entities and dependency structures. In our framework, the depth of the linguistic annotation can vary depending on the domain and the available resources.

The **classifier** has the task of delivering relevant paragraphs and sentences that contain seed elements. It has three subcomponents: *document retrieval*, *paragraph retrieval* and *sentence retrieval*. The document retrieval component utilizes a standard information retrieval system, taking the seed as free text query. A translation step is built in to convert the seed into the proper IR query format. As explained in Xu et al. (2006), all possible lexical variants of the seed arguments are generated to boost the retrieval coverage and formulate a boolean query where the arguments are connected via conjunction and the lexical variants are associated via disjunction. However, the translation could be modified. The task of paragraph retrieval is to find text snippets from the relevant documents where the seed relation arguments co-occur. Given the paragraphs, a sentence containing at least one or two arguments of a seed

relation will be regarded as relevant.

The **rule learning** component is the core component of the ***DARE*** system. It identifies patterns from the annotated documents inducing extraction rules from the patterns, and validates them. In next section, we will give a detailed explanation of this component. The relation extraction component applies the newly learned rules to the relevant documents and extracts relation instances. The validated relation instances will then be used as new seeds for the next iteration.

5.6 Pattern Extraction

Pattern extraction in ***DARE*** aims to find linguistic patterns which not only trigger the relations but also locate the relation arguments and assign the corresponding semantic roles to the arguments. In ***DARE*** the patterns can be extracted from a phrase, a clause or a sentence, depending on the location and the distribution of the seed relation arguments. Given a n-ary relation as the target relation and linguistic analysis of the relevant sentences annotated with the seed relation arguments, we systematically extract patterns which contain one to n arguments of the target relation.

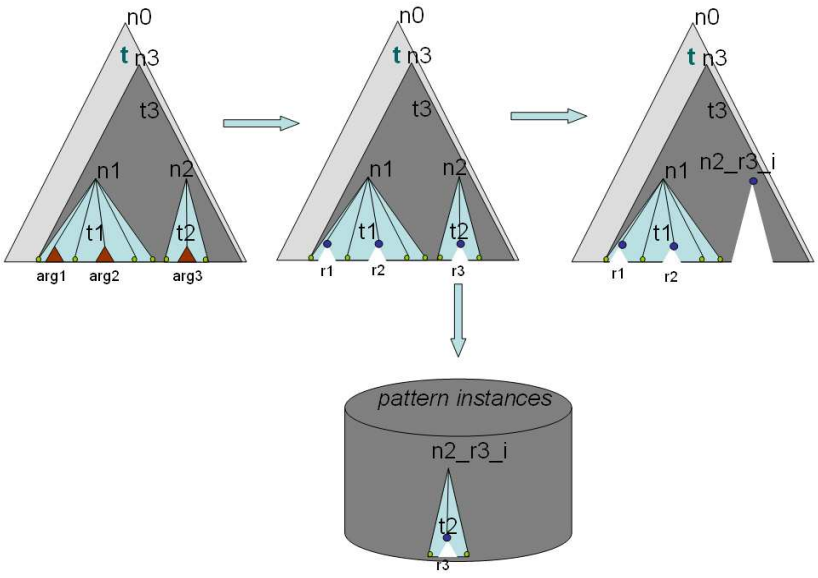

Figure 5.2: Pattern extraction step 1

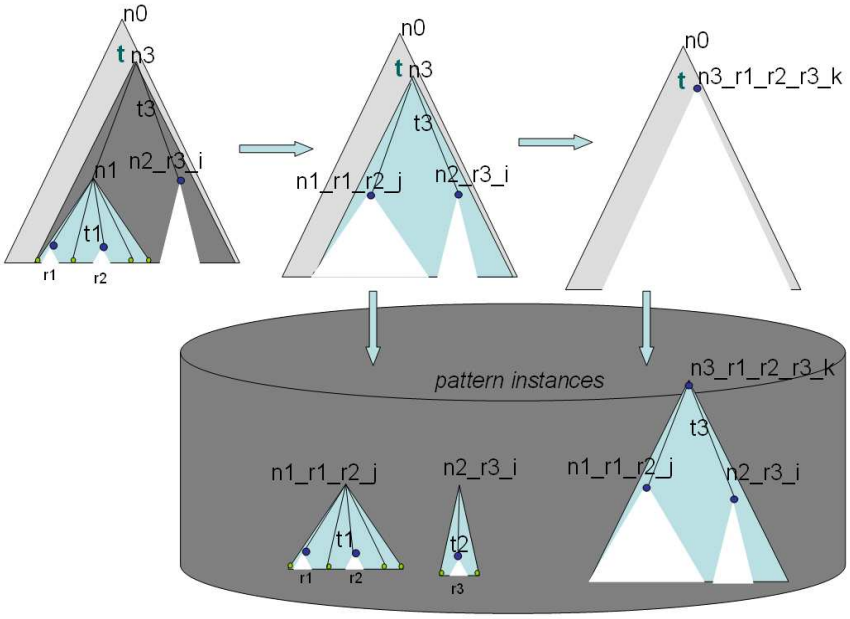

Figure 5.3: Pattern extraction step 2

Figures (5.2) and (5.3) depict the general steps of bottom-up pattern extraction from a dependency tree t where three seed arguments arg_1, arg_2 and arg_3 are located. All arguments are assigned to their relation roles r_1, r_2 and r_3. The pattern-relevant subtrees are trees in which seed arguments are embedded: t_1, t_2 and t_3. Their root nodes are n_1, n_2 and n_3. Figure (5.2) shows the extraction of a unary pattern $n_2 r_3\_i$, while Figure (5.3) illustrates the further extraction and construction of a binary pattern $n_1\_r_1\_r_2\_j$ and a ternary pattern $n_3\_r_1\_r_2\_r_3\_k$. In practice, not all branches in the subtrees will be kept.

In the following, we give a general definition of our seed-driven bottom-up pattern extraction algorithm:

- `input`:
  - $relation = \langle r_1, r_2, ..., r_n \rangle$: the target relation tuple with n argument roles;
  - `T`: a set of linguistic analysis trees annotated with the seed relation arguments (e.g., arg_1, arg_2, ..., arg_n)
- `output`:

`P`: a set of pattern instances which can extract n or a subset of n arguments. The pattern instances are indexed by the argument role combination. For example, $r_1 r_2$ patterns are patterns which can extract arguments r_1 and r_2.

- `Pattern extraction`:
  for each tree `t` $\in$`T`
  - `Step 1:` (depicted in Figure 5.2)
    1. replace all terminal nodes that are instantiated with the seed arguments by new nodes. Label these new nodes with the seed argument roles and the corresponding entity classes;
    2. identify the set of the lowest nonterminal nodes $\texttt{N}_1$ in `t` that may dominate among other nodes at most one argument;
    3. substitute $\texttt{N}_1$ by nodes labelled with the seed argument roles and their entity classes;
    4. prune the subtrees dominated by $\texttt{N}_1$ from `t` and add these subtrees to `P`. These subtrees are assigned with the argument role information and a unique id.
  - `Step2:` **For** i=2 to n: (depicted in Figure 5.3)
    1. find the set of the lowest nodes $\texttt{N}_i$ in `t` that dominate in addition to other children only i seed arguments;
    2. substitute $\texttt{N}_i$ by nodes that are labelled with the i seed argument role combination information (e.g., $r_m r_n$) as well as with a unique id.
    3. prune the subtrees $\texttt{T}_i$ dominated by $\texttt{N}_i$ from `t`;
    4. add $\texttt{T}_i$ together with the argument role combination information and the unique id to `P`

Our pattern extraction algorithm works bottom-up. It discovers patterns for extracting relations with various complexity by allowing the triggering of less complex patterns within a pattern. With this approach, we can learn rules like (5.6) in a straightforward way. In the following, we list two pattern rules in a simplified format:

- [$_{rule}$⟨subject: organization⟩ “appoint” ⟨object: person_in⟩ ⟨infinitive: [$_{rule}$ “succeed”⟨object: person_out⟩⟩]]

- [$_{rule}$⟨subject: person⟩ "name" ⟨object: person_in⟩ ⟨infinitive: [$_{rule}$ "be"⟨pred: position [$_{rule}$ ⟨gen: organization ⟩⟩⟩]]]

5.7 Rule Induction

Given the bottom-up extracted patterns, the task of the rule induction is to reduce the number of patterns to ease the search space of the pattern application. The ***DARE*** rule induction is inspired by the bottom-up rule induction strategy (Califf and Mooney 2004). Based on the specific properties of our patterns and their relationships, the ***DARE*** rule induction carries out three main tasks: rule grouping, redundancy deletion and compression of similar rules. The pattern compression starts bottom-up from one argument pattern to n argument pattern. In the current system, two patterns are similar when

- they extract the same argument role combination,
- their root nodes share the same linguistic annotation, namely, the same head information,
- they contain the same number of daughters,
- their daughters are similar, when
  - they share the same linguistic annotation and/or
  - they trigger various rules that extract the same subset of arguments.

If the similarity conditions are fulfilled, two rules can be compressed into one rule. Let us look at the following two examples in the prize award domain (see (5.8) and (5.9)).

(5.8) Robert Mundell has won the 1999 Nobel Prize for Economics.

(5.9) J.G. Veltman won the 1999 Nobel Prize in physics.

The trees in Figure (5.4) and Figure (5.5) only differ in the usage of the prepositions "in" or "for". We can extract three patterns from each tree: (5.10), (5.11) and (5.12) are from the tree in Figure (5.4), while (5.13), (5.14) and (5.15) are from the tree in Figure (5.5).

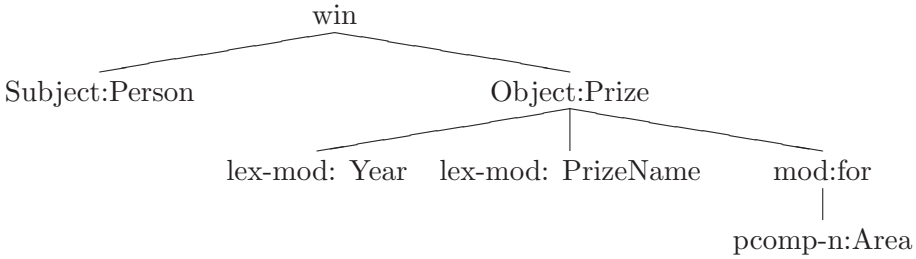

Figure 5.4: Depedency tree analysis of example (5.8)

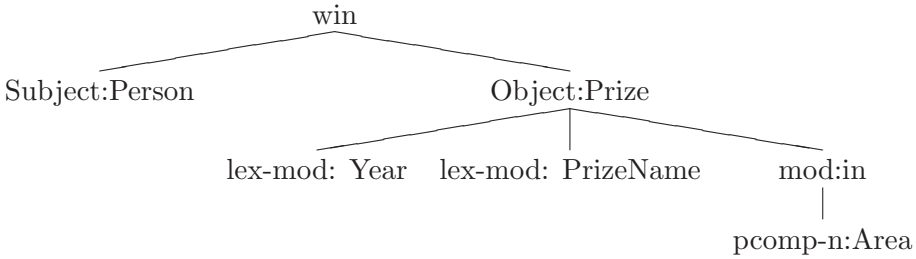

Figure 5.5: Dependency tree analysis of example (5.9)

(5.10) `Rule name::` `area_1`

`Rule body::`
$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{preposition} \\ \text{lex-form} & \text{“for”} \end{bmatrix} \\ \text{daughters} & \langle \begin{bmatrix} \text{pcomp-n} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Area} \end{bmatrix} \end{bmatrix} \rangle \end{bmatrix}$$

`Output::` $\langle \boxed{1} Area \rangle$

(5.11) `Rule name::` `year_prize_area_1`

`Rule body::`
$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{noun} \\ \text{lex-form} & \text{“prize”} \end{bmatrix} \\ \text{daughters} & \langle \begin{bmatrix} \text{lex-mod} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Year} \end{bmatrix} \end{bmatrix}, \\ & \begin{bmatrix} \text{lex-mod} & \begin{bmatrix} \text{head} & \boxed{2}\ \text{Prize} \end{bmatrix} \end{bmatrix}, \\ & \begin{bmatrix} \text{mod} & \begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{preposition} \\ \text{lex-form} & \text{“for”} \end{bmatrix} \\ \text{rule} & \text{area\_1::} \langle \boxed{3} Area \rangle \end{bmatrix} \end{bmatrix} \rangle \end{bmatrix}$$

`Output::` $\langle \boxed{1} Year, \boxed{2} Prize, \boxed{3} Area \rangle$

(5.12) `Rule name::` `recipient_prize_area_year_1`

`Rule body::`

$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{verb} \\ \text{mode} & \text{active} \\ \text{lex-form} & \text{“win”} \end{bmatrix} \\ \text{daughters} & \langle \begin{bmatrix} \text{subject} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Person} \end{bmatrix} \end{bmatrix}, \\ & \begin{bmatrix} \text{object} & \begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{noun} \\ \text{lex-form} & \text{“prize”} \end{bmatrix} \\ \text{rule} & \text{year\_prize\_area\_1::}\ \langle \boxed{4}Year, \boxed{2}Prize, \boxed{3}Area \rangle \end{bmatrix} \end{bmatrix} \rangle \end{bmatrix}$$

`Output::` $\langle \boxed{1}Recipient, \boxed{2}Prize, \boxed{3}Area, \boxed{4}Year \rangle$

(5.13) `Rule name::` `area_2`

`Rule body::`

$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{preposition} \\ \text{lex-form} & \text{“in”} \end{bmatrix} \\ \text{daughters} & \langle \begin{bmatrix} \text{pcomp-n} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Area} \end{bmatrix} \end{bmatrix} \rangle \end{bmatrix}$$

`Output::` $\langle \boxed{1}Area \rangle$

(5.14) `Rule name::` `year_prize_area_2`

`Rule body::`

$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{noun} \\ \text{lex-form} & \text{“prize”} \end{bmatrix} \\ \text{daughters} & \langle \begin{bmatrix} \text{lex-mod} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Year} \end{bmatrix} \end{bmatrix}, \\ & \begin{bmatrix} \text{lex-mod} & \begin{bmatrix} \text{head} & \boxed{2}\ \text{Prize} \end{bmatrix} \end{bmatrix}, \\ & \begin{bmatrix} \text{mod} & \begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{preposition} \\ \text{lex-form} & \text{“in”} \end{bmatrix} \\ \text{rule} & \text{area\_2::}\ \langle \boxed{3}Area \rangle \end{bmatrix} \end{bmatrix} \rangle \end{bmatrix}$$

`Output::` $\langle \boxed{1}Year, \boxed{2}Prize, \boxed{3}Area \rangle$

(5.15) `Rule name::` `recipient_prize_area_year_2`

`Rule body::`

$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{verb} \\ \text{mode} & \text{active} \\ \text{lex-form} & \text{“win”} \end{bmatrix} \\ \text{daughters} & \left\langle \begin{bmatrix} \text{subject} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Person} \end{bmatrix} \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{object} & \begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{noun} \\ \text{lex-form} & \text{“prize”} \end{bmatrix} \\ \text{rule} & \text{year\_prize\_area\_2::}\ \langle \boxed{4}Year, \boxed{2}Prize, \boxed{3}Area \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

`Output::` $\langle \boxed{1}Recipient, \boxed{2}Prize, \boxed{3}Area, \boxed{4}Year \rangle$

Our current induction component will compress rule (5.12) and rule (5.15) together and formulate a new rule (5.16) which triggers the rule year_prize_area_1 and year_prize_area_2 at the object position.

(5.16) `Rule name::` `recipient_prize_area_year_3`

`Rule body::`

$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{verb} \\ \text{mode} & \text{active} \\ \text{lex-form} & \text{“win”} \end{bmatrix} \\ \text{daughters} & \left\langle \begin{bmatrix} \text{subject} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Person} \end{bmatrix} \end{bmatrix}, \right. \\ & \left. \begin{bmatrix} \text{object} & \begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{noun} \\ \text{lex-form} & \text{“prize”} \end{bmatrix} \\ \text{rule} & \text{year\_prize\_area\_1} \mid \text{year\_prize\_area\_2::} \\ & \langle \boxed{4}Year, \boxed{2}Prize, \boxed{3}Area \rangle \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

`Output::` $\langle \boxed{1}Recipient, \boxed{2}Prize, \boxed{3}Area, \boxed{4}Year \rangle$

The algorithm can be described as follows:

The ***DARE*** Rule Induction Algorithm

- `input`:
  `P`: a set of patterns extracted by the pattern extraction component;
- `output`:
  `R`: a set of rules induced from `P`.
- `Rule Induction`:

  (1) `rule grouping`:
  given a n-ary target relation, we sort pattern groups with all argument role combinations from r_i $(1 < i < n)$ to $r_1 r_2 ... r_{n-1} r_n$;

  (2) `duplicate deletion and rule compression`— compress similar rules (`compress(P)`) and remove duplicate rules (`deletionDuplicate(P)`):

```
the iteration loop
R=P
R_last=∅
while |R|≠|R_last|
      R_last=R
      R=compress(R) //the rule set after compression
      R=deletionDuplicate(R) //the rule set after compression
end while
return R


deletionDuplicate(P)
for i = 0 to |P|-1 do
    for j = i + 1 to |P| do
        p_i is the ith pattern in P
        p_j is the jth pattern in P
        if p_i = p_j then
           P = P - p_j // delete p_j from P
           P=P(p_j/p_i)// substitute all occurrences of p_j by p_i
        end if
    end for
end for
return P;
```

```
compression(P)
for i = 0 to |P|-1 do
    for j = i + 1 to |P| do
        p_i is the ith pattern in P
        p_j is the jth pattern in P
        p_i contains a head node n_1 and a list of daughter nodes D_1 and
        p_j contains a head node n_2 and a list of daughter nodes D_2
        set p_3 as a new tree
        set D_3 as a new list
        if p_i.output = p_j.output and n_1 = n_2 and |D_1| = |D_2| then
            p_3.head=n_1
            |D_3| = |D_1|
            size = |D_1|
            for m = 0 to size do
                set node_3 as a new tree
                node_1 = D_1[m]
                node_2 = D_2[m]
                if node_1 = node_2 then
                    node_3=node_1
                end if
                if node_1.rule.output = node_2.rule.output and
                    node_1.head = node_2.head then
                    rule_1=node_1.rule
                    rule_2=node_2.rule
                    node_3.rule=(rule_1|rule_2)
                    node_3.head=node_1.head
                end if
            D_3[m] = node_3
            end for
            p_3.daughters=D_3
            P = P - p_1 - p_2 // delete p_1 and p_2 from P
            P=P(p_2/p_3, p_1/p_3 )// substitute all occurrences of p_1 and p_2 by p_3
        end if
    end for
end for
return P;
```

The current algorithm requires equality between the heads of the two nodes. A relaxation of the exact match can be that two heads are similar when their lex-

forms belong to the same concept or share the same upper concept in the domain ontology. For example, "award" and "prize" are synonyms in the WordNet (Miller et al. 1998). If the lex-form in the object in rule (5.14) is "award" instead of "prize" and we allow the compression at the lexical semantic level, the compressed rule will take the following form:

(5.17) `Rule name:: recipient_prize_area_year_3`

`Rule body::`

$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{verb} \\ \text{mode} & \text{active} \\ \text{lex-form} & \text{“win”} \end{bmatrix} \\ \text{daughters} & \langle \begin{bmatrix} \text{subject} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Person} \end{bmatrix} \end{bmatrix}, \\ & \begin{bmatrix} \text{object} & \begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{noun} \\ \text{lex-form} & \text{“prize”} \mid \text{“award”} \\ \text{sense-id} & \text{“10001”} \end{bmatrix} \\ \text{rule} & \text{year\_prize\_area\_1} \mid \text{year\_prize\_area\_2::} \\ & \langle \boxed{4}Year, \boxed{2}Prize, \boxed{3}Area \rangle \end{bmatrix} \end{bmatrix} \rangle \end{bmatrix}$$

`Output::` $\langle \boxed{1}Recipient, \boxed{2}Prize, \boxed{3}Area, \boxed{4}Year \rangle$

In (5.17), we introduce a feature "sense-id" which refers to the concept identifier in a domain ontology or the sense identifier in a lexical semantic ontology. Rule (5.17) is much more general than rule (5.16), since its object can match all words belonging to the concept or the sense with the identifier "10001".

5.8 Ranking and Validation

The ***DARE*** ranking strategy incorporates the ideas proposed by Riloff (1996), Agichtein and Gravano (2000) and Yangarber (2001). It take two properties of a pattern into account:

- domain relevance: its distribution in the relevant documents and irrelevant documents (documents in other domains)
- trustworthiness of its origin: the relevance score of the seeds from which it is extracted.

In Riloff (1996) and Sudo et al. (2003), the relevance of a pattern is mainly dependent on its occurrences in the relevant documents with respect to the whole corpus. Patterns which exhibit low occurrence frequency but are nevertheless relevant cannot float to the top. It is known that some complex patterns have low occurrence but are very relevant. A new method is proposed to calculate the domain relevance of a pattern. It is assumed that the domain relevance of a pattern is dependent on the relevance of the terms constructing the pattern.

5.8.1 Domain Relevance Score

Given n completely different domains, the **domain relevance score** (DR) of a term t in a domain d_i is:

$$DR(t,d_i) = \begin{cases} 0 & \text{if } df(t,d_i)=0, \\ \frac{df(t,d_i)}{D \times N} \times Log(n \times \frac{df(t,d_i)}{\sum_{j=1}^{n} df(t,d_j)}) & \text{otherwise.} \end{cases} \tag{5.18}$$

where

- $df(t,d_i)$: is the document frequency of a term t in the domain d_i
- D: is the number of documents in the domain
- N: is the total number of terms in the domain
- n: is the number of the domains

Here the domain relevance of a term is dependent both on its document frequency and its document frequency distribution in other domains. Terms mentioned by more documents within the domain than outside are more relevant (Xu et al. 2002). In the case of n=3 such different domains might be, e.g., management succession, book review or biomedical texts. Every domain corpus should ideally have the same number of documents and roughly similar average document size. In Section 4.2, a detailed explanation of our approach to learning domain relevant terms is given.

5.8.2 Relevance Score of a Pattern

In the calculation of the **trustworthiness** of the origin, we follow the basic idea of Agichtein and Gravano (2000) and Yangarber (2001). We take the value of the most trustworthy seed from which a pattern is extracted as the trustworthiness value of this pattern. Thus, the relevance of a pattern in our system is dependent on the relevance of its terms and its trustworthiness value. Finally, the **relevance score** of a pattern p is calculated as follows:

$$score(p) = \begin{cases} \sum_{i=0}^{|T|} DR(t_i) \times max\{score(s) : s \in Seeds\} & \text{if } |T| > 0, \\ max\{score(s) : s \in Seeds\} \times c & \text{if } |T| = 0. \end{cases} \tag{5.19}$$

where $t_i \in T$ and $0 \leq score(p) < 1$

- T: is the set of the terms occurring in p;
- $Seeds$: is a set of seeds from which the pattern is extracted
- $score(s)$: is the score of the seed s
- c: is the highest rank of the domain relevant terms

This relevance score is not dependent on the distribution frequency of a pattern in the domain corpus. Therefore, patterns with lower frequency, in particular some complex patterns, can be ranked higher when they contain relevant domain terms and are from the reliable seeds.

5.8.3 Relevance Score of a Seed

According to the duality principle (Yangarber 2001), the score of the newly extracted tuple *Tuple* is dependent on the patterns from which it originates. Our scoring method is a simplified version of that defined by Agichtein and Gravano (2000):

$$score(Tuple) = 1 - \prod_{0}^{|p|}(1 - score(P_i)) \tag{5.20}$$

where $P = \{P_i\}$ is the set of patterns that extract the *tuple*. The extracted tuples can be used as potential seeds for the pattern extraction. The initial seeds are assigned 1 as their score.

5.9 Top Down Rule Application

After the acquisition of pattern rules, the ***DARE*** system applies the validated ones to the linguistically annotated corpus to extract additional relation instances[1]. All sentences in the annotated corpus have been analyzed by the named entity recognition and the dependency parser. The entity information is marked up in the dependency trees. In order to achieve good coverage, referential expressions share the same named entity information with their antecedents, e.g.,

(5.21) *Wiesel*$_1$, *who*$_1$ won the Nobel Peace Prize in 1986, stopped *Barije Redinica*$_2$, 16, as *she*$_2$ walked by in the camp.

The relative pronoun "who" in (5.21) refers to the person name "Wiesel", while the pronoun "she" refers to the person name "Barije Redinica". As soon as the co-reference relationship is identified, we add the entity information to the referential expressions.

We decide on a top-down rule application strategy where complex rules are preferred over simpler ones, namely, applying the most complex patterns to the analyzed sentence in order to extract the maximal number of the relation arguments. However, patterns with the same complexity can match one tree structure, although they extract different argument combinations. In the following, we give some examples.

(5.22) *Aung San Suu Kyi*, 56, was awarded the *Nobel Peace* Prize in *1991*.

Two rules can be applied to the sentence (5.22). Both rules can extract three arguments.

[1]Examples in this section are provided by Li (2006).

(5.23) **rule5:** ⟨ *nobel*, *peace*, *1991*, [ ] ⟩

rule26: ⟨ *nobel*, *peace*, [ ], *Aung San Suu Kyi* ⟩

Given these alternative extracted partial results (projections of a relation instance), the unification will only apply to them if the two partial tuples share at least one argument. This additional constraint is also proposed by McDonald et al. (2005) for combining partial results to a complex relation instance. The two extracted partial results in (5.23) can be merged to the following event instance:

(5.24) ⟨ *nobel*, *peace*, *1991*, *Aung San Suu Kyi* ⟩

If the unification fails, we will keep the partial results and let the validation component make the decision, see (5.25).

(5.25) For his efforts, *Trimble* has been lauded internationally, sharing the *Nobel Peace* Prize with *John Hume*, the pacifist nationalist leader.

rule6: ⟨ *nobel*, *peace*, [ ], *Trimble* ⟩

rule13: ⟨ *nobel*, *peace*, [ ], *John Hume* ⟩

5.10 Conclusion

In this chapter, we have described a framework for minimally supervised learning of patterns for relation extraction from text, which is an extension and elaboration of the work presented in Xu et al. (2007). This framework follows in the tradition of Riloff (1996), Brin (1998), Agichtein and Gravano (2000) and Yangarber (2001), who have proposed various ways to learn relation extraction patterns from texts with a minimal amount of seed knowledge. The seed-based bootstrapping approaches are theoretically attractive because the learned patterns and rules are modular and transparent. They can be reused in new applications and they can be a valuable resource for (computational) linguistic investigation. The learning algorithms are not domain dependent.

The novelties of our approach are that

- it can learn relations of any arity,
- it attaches semantic role labels to the extracted arguments (so that a subsequent mapping process from extraction-pattern to IE-template is avoided),
- it can learn patterns not just from subject-verb-object triples but from any dependency structures,
- the rules learned can be recursive (in the sense that a complex rule can contain an embedded simpler rule),
- the rule induction strategy can compress the rules not only bottom-up but also recursively.

Like other bootstrapping approaches such as Agichtein and Gravano's Snowball system, we start with seed instances of a relation and retrieve text snippets in which the seed instance arguments co-occur. From linguistically annotated versions of these snippets, patterns are learned, extraction rules induced and then validated.

We use a rich attribute value matrix (AVM) rule representation formalism that allows the association of semantic roles with elements of the dependency structure found in linguistic annotation and also allows embedded AVM structures enabling rules to contain subrules, which may therefore be reused in multiple contexts.

Extracted patterns are scored based on their domain relevance (distribution in relevant and irrelevant documents) and the trustworthiness of the seeds on which it is based. Accepted patterns are used to retrieve more seed instances which themselves are scored based on the scores of the patterns from which they originate. A new rule induction method is developed to compress rules in a bottom-up strategy: simple rules first and then the more complex rules.

In the next chapter, we will present the experiments of the ***DARE*** framework and the corresponding evaluations.

Chapter 6

Experiments and Evaluation

We apply the ***DARE*** framework to two domains: prize award and management succession events. For the seed construction, a study is carried out to estimate the effectiveness of seed relations of different arity to see how much they can contribute to the learning of successful patterns. Evaluations are conducted to investigate the ***DARE*** system performance (precision and recall) with respect to the seed parameters: the number of seeds, the influence of data size and its redundancy property. Performance for the Nobel Prize task turns out to be especially promising. For the management succession task, the results compare favorably with those of existing pattern acquisition approaches. Furthermore, an investigation of the differences in behavior between the Nobel Prize award and management succession demonstrate that size and properties of the data play an important role in the success of the ***DARE*** method. A detailed analysis of learning and extraction performance during bootstrapping is presented for the Nobel Prize task. The error analysis identifies various sources of incorrect instance detection. A special analysis is dedicated to monitoring the error spreading problem in the bootstrapping process, with respect to the interplay between seeds and rules. Finally, three extended scenarios are constructed to test the enticing idea of reusing rules learned from a benevolent data set in one domain to domains lacking the desired degree of redundancy in their data.

6.1 Experimental Domains and Data Resources

We have started with the Nobel Prize award domain since it is a domain for which complete records of all awarded prizes can be obtained in structured formats and in addition a large number of free texts about awards and laureates can be found on the web. Furthermore the data are manageable in size, authoritative and can be used for the creation of a gold-standard for seed selection and evaluation. Table 6.1 presents an overview of our test data sets.

Data Set Name	Document Number	Data Amount
Nobel Prize **A** (1999-2005)	2296	12,6 MB
Nobel Prize **B** (1981-1998)	1032	5,8 MB
Nobel Prize **A+B** (1981-2005)	3328	18,4 MB
MUC-6	199	1 MB

Table 6.1: Overview of test data sets

For the Nobel Prize award scenario, we divide the Nobel Prize corpus into three parts: the Nobel Prize A, the Nobel Prize B and their combination (the total corpus). The texts in the corpus are Nobel Prize related articles from New York Times, online BBC and CNN news reports:

- texts from New York Times from June 1998 to September 2000 (part of the AQUAINT data)
- online news texts from BBC (November 1997 to December 2005), CNN (October 1995 to January 2006), New York Times (October 1981 to January 2006)
- reports from Nobel e-Museum[1]

The Nobel Prize A contains data from 1999 to 2005, while the data in the Nobel Prize B are newspapers from 1981 to 1998, almost half the size of the data set A.

We have collected the complete Nobel Prize winner list from the Nobel e-Museum and store it in a relational database. The list contains the following information about the winner: the name, the gender, the award year, the monetary amount of the prize, the position, the affiliation, country, nationality, the prize area. The target relation for the experiment is a quaternary relation:

[1] `http://www.nobel.se/`

⟨ recipient, prize, area, year ⟩

For data sets in this domain, we are faced with an evaluation challenge pointed out by Brin (1998) and Agichtein and Gravano (2000), because there is no gold-standard evaluation corpus available. We have adapted the evaluation method suggested by the Snowball system (Agichtein and Gravano 2000), namely, the Ideal table method (see Section 3.1.3).

For the management succession scenario, we use the test data from MUC-6 (MUC-6 1995) and define a simpler relation structure than the MUC-6 scenario template with four arguments, since we want to compare results between the two domains:

⟨ personIn, personOut, position, organisation ⟩

- personIn: the person who obtained the position
- personOut: the person who left the position
- position: the position that the two persons are involved in
- organisation: the organisation where the position is located

The MUC-6 corpus for the management succession domain is much smaller than the Nobel Prize corpus. Since a gold-standard of the target relations is available, we use the standard IE precision and recall method.

In our experiments, we attempt to investigate the influence of the target relation properties w.r.t the seed behavior, the size of the seed and the size of the test data on the performance. All these documents are processed by named entity recognition (Drożdżyński et al. 2004) and the dependency parser MINIPAR (Lin 1998).

6.2 Tools

The ***DARE*** system contains three main components: the *linguistic annotation*, the *classifier* and the *rule learning* component. *Rule learning* is the core

component of the ***DARE*** system and was described in detail in the previous chapter (see Chapter 5). The *linguistic annotation* is responsible for enriching the natural language texts with linguistic information such as named entities and dependency structures, while the *classifier* has the task of finding relevant paragraphs and sentences that contain seed elements. In general, the *linguistic annotation* component and the *classifier* component are not restricted to utilizing any particular systems and therefore could also integrate any other tools providing the required functionality. In the current ***DARE*** implementation, we have employed an open source search engine *Lucene* for document and paragraph retrieval, the *SProUT* system for named entity recognition (Drożdżyński et al. 2004), and the dependency parser MINIPAR (Lin 1998) for sentence structure analysis.

6.2.1 Lucene

Lucene[2] is a well-known open source full text search engine, written entirely in the Java programming language. It provides efficient batch indexing, fast storing, and powerful searching capabilities. A well-defined API enables the developers to implement their application specific indexing and search functionalities in a very convenient way. Various search functionalities are available in *Lucene*, e.g.,

- ranked searching (the most relevant documents are returned first)
- different query types: standard boolean query, phrase queries, wildcard queries, etc.
- typed search (e.g., search terms can be typed as for example, *title*, *person* or *company*)
- sorting by any type

The typed search is a very useful function, since it helps find text fragments containing the terms belonging to certain semantic concepts. A text preprocessed by a named entity recognition system can deliver the input for a typed search indexing. A typed search query looks like the following:

[2] `http://lucene.apache.org`

(6.1) ⟨ **prize**: Nobel⟩ `AND` ⟨**word**: win ⟩

Lucene will find all documents or paragraphs containing the word *win* and the prize name *Nobel*. Thus, a ***DARE*** semantic seed can be translated as a typed search query:

(6.2) Seed: ⟨ "*Ahmed H. Zewail*"|"*Ahmed Zewail*"|"*Zewail*",
"*Nobel*",
"*Chemistry*"|"*Chemist*"|"*Chemical*",
"*1999*" ⟩

Lucene Query: (⟨**person**: Ahmed H. Zewail ⟩ `OR`
⟨**person**: Zewail⟩ `OR`
⟨**person**: Ahmed Zewail⟩) `AND`
⟨**prize**: Nobel⟩ `AND`
(⟨**area**: Chemistry⟩ `OR` ⟨**area**: Chemist⟩) `AND`
⟨**year**: 1999⟩

In the above example, we have also included the lexical variants of person or area names. In the current experiment, we apply *Lucene* as a document retrieval tool both for the *classifier* component and for the general data collection task.

6.2.2 *SProUT*

SProUT is a platform for developing multilingual Shallow Text Processing and IE systems. The entire system has been developed in Java. In Section 4.4, we have described the *SProUT* system, in particular the usage of its XTDL formalism for rule definition. *SProUT* offers a very user-friendly grammar development environment due to its elegant grammar formalism. We have extended the existing *SProUT* general named entity classes (person, organization, date time, currency) with new classes such as prize name and the area name for our application domain.

6.2.3 MINIPAR

We apply MINIPAR (Lin 1998) to our corpus to obtain dependency structures[3]. We selected MINIPAR among other powerful free dependency parsers because of its efficiency, robustness and rich structures. Furthermore, MINIPAR is particularly robust for dealing with texts such as online texts which also contain fragmented and not well-formed sentences. Thus, it is widely used by other IE systems too (e.g., Jijkoun et al. (2004), Stevenson and Greenwood (2005), Stevenson and Greenwood (2006) and Romano et al. (2006)). However, our selection is not based on a systematic comparative evaluation. The close contenders of MINIPAR could have been utilized as well. We will integrate other parsers in future work, such as the Stanford parser (Klein and Manning 2003). As evaluated by Stevenson and Greenwood (2006), the Stanford parser has a better coverage for the linked chain model than MINIPAR.

MINIPAR is a principle-based, broad-coverage parser for English. The grammar representation is a network of nodes and links, where the nodes are grammatical categories and the links are types of dependency relationships, such as subject, object and modifier. MINIPAR takes one sentence as an input and determines the dependency relationships among the words. To deal with parse ambiguities, MINIPAR makes use of the frequency counts of the grammatical dependency relationships extracted by a collocation extractor from a 1GB corpus. The dependency tree with the highest ranking is returned as the parse of the sentence. The MINIPAR lexicon contains about 130,000 entries, derived from WordNet with additional proper names. The lexicon entry of a word lists all applicable parts of speech of the word and its subcategorization frames, if these exisit. MINIPAR achieves about 88% precision and 80% recall with respect to dependency relationships, evaluated on the syntactically annotated Susanne corpus, a subset of the Brown Corpus of American English.

A special module was developed for ***DARE*** that combines the named entity recognition results with the dependency structure analysis. Li (2006) provides a detailed description of the usage of the NLP annotation modules.

[3] `http://www.cs.ualberta.ca/~lindek/minipar.htm`

6.3 Seed Behavior

We conducted a series of experiments with the tasks of investigating the behavior of the seed complexity and its influence on the relevant sentence retrieval for the Nobel Prize winning event (Xu et al. 2006) and the management succession event. In our experiment, we start from the entire list of Nobel Prize winners of 1998 and 1999. Our Nobel-Prize winning event seed contains four arguments: *recipient*, *prize name*, *year* and *area*:

$$(6.3)\quad \begin{bmatrix} \textbf{recipient} & \text{person or organization} \\ \textbf{prize} & \text{prizename} \\ \textbf{area} & \text{area} \\ \textbf{year} & \text{year} \end{bmatrix}$$

Since the seed is a semantic relation, we can also map any slot value to a number of patterns. Thus, we have generated all variants of the potential mentions of person names or areas, in order to boost the matching coverage of our seeds with the texts. For example, for the person name, *Alan J. Heeger*, its mentions can be *Alan J. Heeger*, *Alan Heeger*, *Heeger*, and *A. J. Heeger*. We did the same with the prize area, e.g., the mention variants of *Chemistry* can be *chemical*, sometimes the professional description *chemist* provides also an indication of the area. Then a seed instance looks as follows:

$$\begin{bmatrix} \textbf{recipient} & \text{“Alan Heeger” | “Alan J. Heeger” | “A. J. Heeger”| “Heeger”} \\ \textbf{prize} & \text{“Nobel”} \\ \textbf{area} & \text{“Chemistry” | “chemical” | “chemist”} \\ \textbf{year} & \text{“2000”} \end{bmatrix}$$

Thus, we annotated our training texts in the Nobel Prize Domain with the entity mentions of the seed events automatically, using *SProUT*. Then all sentences containing entity mentions of the seeds are extracted by our system. The extracted sentences are sorted by the number of event arguments contained: quaternary, ternary and binary complexity. A sentence with quaternary complexity is a sentence containing all four arguments of one event seed. Within ternary complexity and binary complexity, we classify them into different groups according to the entity class combination, e.g., $< person, area, time >$, $< person, prize, area >$, $< person, area >$, etc. Then we evaluated whether these

sentences are about the Nobel Prize winning event. In Table 6.2, we show the distribution of the seed complexity in the sentences describing the events.

complexity	matched sentence	relevant event extent	precision %
4-ary	36	34	94.0
3-ary	110	96	87.0
2-ary	495	18	3.6

Table 6.2: Nobel Prize domain: distribution of the seed complexity

For the entity-class combinations, e.g., 3-ary and 2-ary, the projections of the target relation, we also carried out a distribution count, presented in Table 6.3.

combination (3-ary, 2-ary)	matched sentence	relevant event extent	precision %
person, prize, area	103	91	82.0
person, prize, time	0	0	0.0
person, area, year	1	1	100.0
prize, area, year	6	4	68.0
person, prize	40	15	37.5
person, area	123	0	0.0
person, year	8	3	37.5
prize, area	286	0	0.0
prize, year	25	0	0.0
area, year	12	0	0.0

Table 6.3: Nobel Prize domain: distribution of relation projections

Table 6.2 tells us that the more event arguments a sentence contains, the higher the probability is that the sentence is an event extent. Table 6.3 shows the difference between different entity class combinations with respect to the event identification. We can potentially regard these values as additional validation criteria for event extraction rules. Whereas Table 6.2 helps us preestimate the contribution of the different arity classes for successful event extraction, Table 6.3 shows us which types of incomplete seeds might be most useful. Both distributions, especially the second one, will be very much dependent on the kind of relations to be extracted. Such seed analyses could be used to better characterize a given relation-extraction task.

The target relation in the management succession domain is a little more problematic than the target Nobel Prize award relation, since the same entity concept person can assume the role either of personIn or personOut. We constructed two relation instance sets for the evaluation of the seed behavior. The relation instances are extracted from the gold-standard annotation.

- *ambiguous set*: a set of relation instances where the same person in the same corpus occurs also as personIn in a relation instance and has the personOut role in another relation instance.
- *unambiguous set*: a set of relation instances where a person has only personIn or just personOut role in the corpus.

There are 60 instances occurring in the corpus belonging to the ambiguous set, while 55 instances belong to the unambiguous set. At first, we put the two sets of instances together and calculated the general distribution of the seed complexity.

complexity	matched sentence	relevant event extent	precision %
4-ary	21	19	90.4
3-ary	102	77	75.4
2-ary	206	86	40.7

Table 6.4: Management succession: distribution of the seed complexity

Table 6.4 confirms our interpretation of Table 6.2 that the greater the arity of the seed relation, the higher the precision of relevant sentence retrieval. However, both tables also show that the less complex projections of the target relation help find more relevant sentences. Therefore, the relation projections play an important role for the improvement of the recall value. For the entity class combinations of 3-ary and 2-ary, we also carried out a distribution count for the two different seed sets, presented in Table 6.5 and Table 6.6.

combination (3-ary, 2-ary)	matched sentence	relevant event extent	precision %
personIn, personOut,organization	6	6	100.0
personIn, personOut,position	10	7	70.0
personIn, organization,position	26	20	76.9
personOut, organization,position	13	9	69.2
personIn, personOut	12	11	91.7
personIn, organization	40	11	27.5
personIn, position	19	8	42.1
personOut, organization	25	4	16.0
personOut, position	6	2	33.3
organization, position	0	0	0.0

Table 6.5: Ambiguous set: distribution of relation projections

If we ignore the combination cases in the unambiguous set where no matches are found, the results of the two entity combinations in Table 6.6 are in general

combination (3-ary, 2-ary)	matched sentence	relevant event extent	precision %
personIn, personOut,organization	8	4	50.0
personIn, personOut,position	12	8	66.7
personIn, organization,position	15	15	100.0
personOut, organization,position	12	8	66.7
personIn, personOut	21	11	52.4
personIn, organization	11	0	0.0
personIn, position	16	9	56.3
personOut, organization	14	8	57.1
personOut, position	8	6	75.0
organization, position	0	0	0.0

Table 6.6: Unambiguous set: distribution of relation projections

much better than those in Table 6.5. This means that the unambiguous relation instances are better seed candidates than the ambiguous relation instances for finding the relevant event extents. Furthermore, we also compared the projections containing both personIn and personOut with the projections containing only one person role, either personIn or personOut. It turns out that the projections with two person roles on the average achieve better precision (73.3%) than the projections with only one person role (48.7%). This gives us a very useful insight into the domain and confirms our discussion about the ambiguous seed example in section 5.3 of the previous chapter:

> A relation instance whose arguments play unambiguous semantic roles in the corpus or which is unambiguous is a better seed candidate for learning unambiguous patterns than relation instances which have potential ambiguities.

An interesting side effect of this study is the observation that there is almost no sentence in the corpus containing only the argument pair *organization* and *position*.

Seed construction analysis helps us learn the characteristics of a relation, and its projections and potential influence on the pattern extraction quality.

6.4 *DARE* Performance

In this section, we evaluate the ***DARE*** system with respect to the interaction between the number of seed relation instances and the data redundancy. Most of the results have already been reported in Xu et al. (2007).

6.4.1 Nobel Prize Award Domain

For this domain, four test runs have been evaluated, initialized each time by one randomly selected relation instance as seed each time. In the first run, we use the second largest test data set *Nobel Prize A*. In the second and third runs, we compare two randomly selected seed samples with 50% of the data each, namely *Nobel Prize B*[4]. The fourth run takes the same seed sample as the first run and applies it to the whole corpus, namely the combination of A and B.

As mentioned above, for data sets in this domain, we are faced with an evaluation challenge pointed out by Brin (1998) and Snowball (Agichtein and Gravano 2000), namely, that no gold-standard evaluation corpus is available. We adapt the evaluation method suggested by Agichtein and Gravano (2000). I.e., our system is successful if we capture one mention of a Nobel Prize winner event through one instance of the relation tuple or its projections.

We construct three Ideal tables reflecting an approximation of the maximal detectable relation instances: one for Nobel Prize A, one for Nobel Prize B and one for their combination. The Ideal tables contain the Nobel Prize winners that co-occur with the word "Nobel" in the test corpus. Since we have the complete list of the Nobel Prize winners, we do not have to construct a *join* table as needed in the Snowball system. Then precision is the correctness of the extracted relation instances, while recall is the coverage of the extracted tuples that match the Ideal table. In Table 6.7 we show the precision and the recall of the four runs and their random seed sample.

All four experiments achieve promising precision values. A significant positive correlation between data size and recall is observed. Corpus A+B has achieved the highest recall, while corpus A has much higher recall than corpus B. All

[4]Some of the initial evaluation results of the Nobel Prize A and B were also reported in Li (2006).

data set	seed	precision %	recall % (total)	recall % (report years)
Nobel Prize A (1999–2005)	⟨[Zewail, Ahmed H], nobel, chemistry, 1999⟩	71.6	50.7	70.9
Nobel Prize B (1981–1998)	⟨[Sen, Amartya], nobel, economics, 1998⟩	87.3	31.0	43.0
Nobel Prize B (1981–1998)	⟨[Arias, Oscar], nobel, peace, 1987⟩	83.8	32.0	45.0
A+B (1981–2005)	⟨[Zewail, Ahmed H], nobel, chemistry, 1999⟩	80.59	62.9	69.0

Table 6.7: Nobel Prize domain: precision, recall against the Ideal Table

four experiments exhibit better recall values when taking into account only the relation instances during the report years, because there are more mentions during these years in the corpus.

The two experiments with the Nobel Prize B corpus show similar performance. Their results tell us that the seed choice in the Nobel Prize award domain is not a crucial issue, at least not for the seeds that were tested, since all Nobel Prize awards are mentioned in the newspaper texts. A statistical investigation of the test corpus shows that some Nobel Prize categories such as the peace and the literature prizes get more news coverage, i.e., have more mentions than the others (Li 2006). However, it is interesting to observe that the linguistic expressions for even less mentioned areas such as Chemistry are general enough for discovery of other event instances.

Figure 6.1 depicts the pattern learning and the new seed extracting behavior during the iterations for the first experiment. Similar behavior is observed in experiments 2, 3 and 4 (see Figure 6.2 and 6.3). That is, the growth of the seed number is almost synchronous with the growth of the rule number: increasing until they reach a peak after two to four iterations, from that point on decreasing until no more rules or seeds can be found. Run 1 and run 4 with larger corpora show much smoother and more harmonized curves than the runs for the smaller corpus B, i.e., runs 2 and 3.

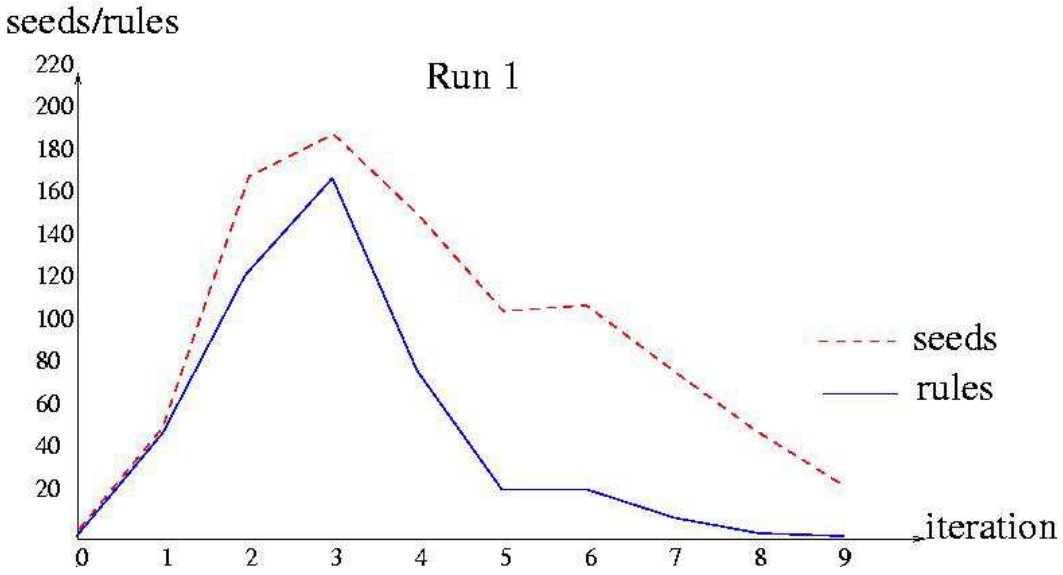

Figure 6.1: Iteration process of run 1 (Nobel Prize A)

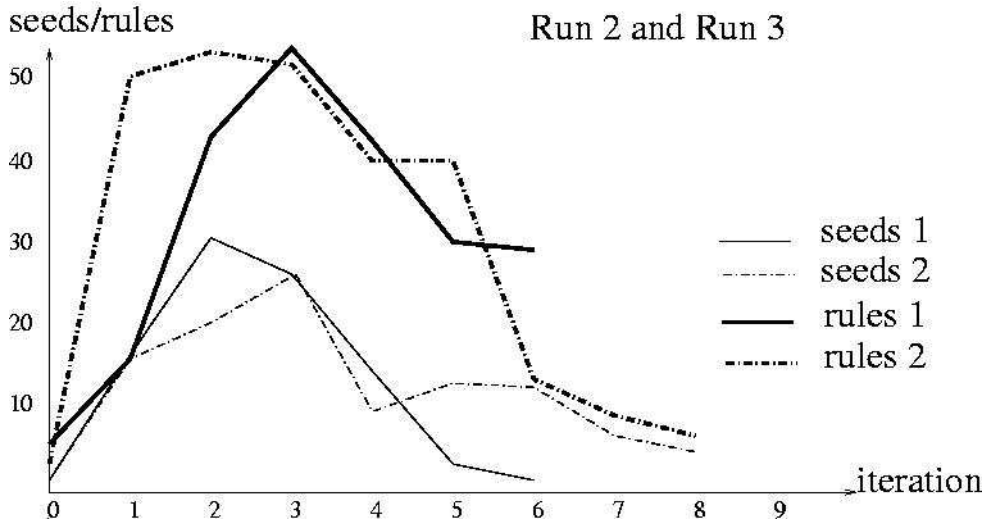

Figure 6.2: Iteration process of run 2 and 3 (Nobel Prize B)

6.4.2 Management Succession Domain

The MUC-6 corpus is much smaller than the Nobel Prize corpus. Since the gold-standard of the target relations is available, we use the standard IE precision and recall method. The total gold-standard table contains 256 event instances, from which we randomly select seeds for our experiments. Table 6.8 presents

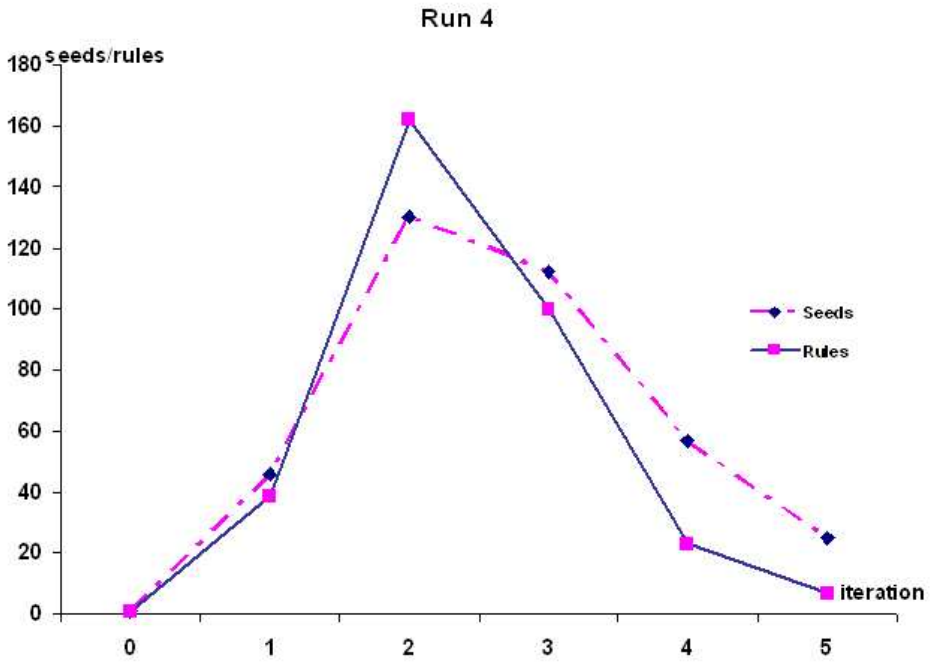

Figure 6.3: Iteration process of run 4 (Nobel Prize A+B)

initial seed nr.	precision %	recall %
1 (a)	12.6	7.0
1 (b)	15.1	21.8
20	48.4	34.2
55	62.0	48.0

Table 6.8: Management succession domain: precision and recall

an overview of the experiment performances. Our tests vary between one seed, 20 seeds and 55 seeds. Some of the results have already been reported in Xu et al. (2007).

The first two tests, which used one seed, achieved poor performance. With 55 seeds, we can extract additional 67 instances to obtain in total roughly 50% of the instances occurring in the corpus. Table 6.9 shows the evaluations w.r.t. individual argument slots. 1(b) works a little better because the randomly selected single seed appears to be a better sample for finding patterns for extracting the PersonIn argument.

Figure 6.4 illustrates the iteration behavior of 1(a) and 1(b). 1(a) has learned

argument	precision % 1 (a)	precision % 1 (b)	recall % 1 (a)	recall % 1 (b)
personIn	10.9	15.1	8.6	34.4
personOut	28.6	–	2.3	2.3
organization	25.6	100	2.6	2.6
position	11.2	11.2	5.5	5.5

Table 6.9: Management succession domain: evaluation of one-seed tests 1(a) and 1(b)

and extracted a very small number of patterns and rules within four iterations. 1(b) has obtained a more synchronous development curve between the patterns and the seeds. As explained above, the good pattern in 1(b) led to the discovery of a large number of new instances filling the personIn argument, therefore, resulting in the steep increase of the seed curve.

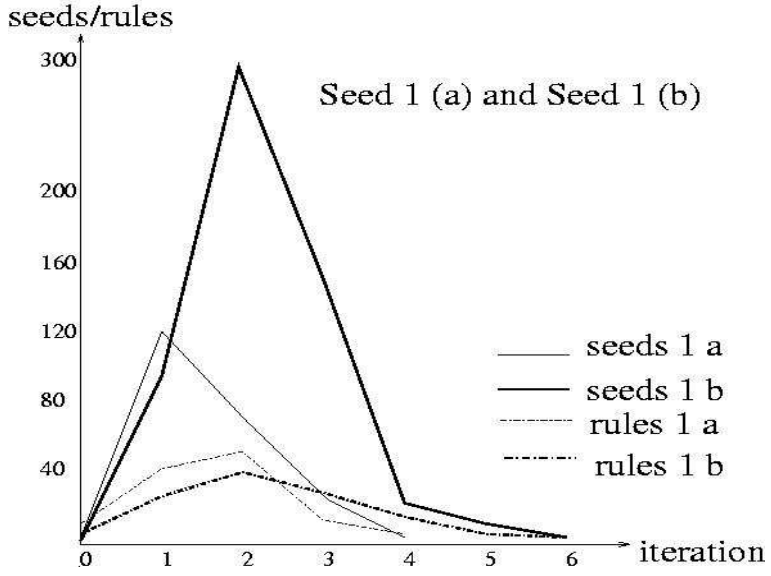

Figure 6.4: Iteration process of run 1(a) and 1(b) (one seed)

Table 6.10 illustrates the performance with 20 and 55 seeds, respectively. Both of them are better than the one-seed tests, while 55 seeds deliver the best average performance, in particular for the recall value.

Figure 6.5 depicts the iteration development of 20 and 50 seed experiments. The iteration curve of 20 seeds is very irregular. The second peak of the seed line implies that pattern rules detected later in the third iteration triggered a

argument	precision % (20)	precision % (55)	recall % (20)	recall % (55)
personIn	84.0	62.8	27.9	56.1
personOut	41.2	59.0	34.2	31.2
organization	82.4	58.2	7.4	20.2
position	42.0	64.8	25.6	30.6

Table 6.10: Management succession domain: evaluation of 20 and 55 seed instances

boost of instance detections. This delays the termination of the process.

The iteration process of the 55 seeds presents a very harmonized interplay between the patterns and the seeds. It is interesting to observe that the whole learning and extraction process ends after only three iterations. The 55 seeds soon detect all additional accessible patterns and the patterns found in one or two iterations all accessible instances.

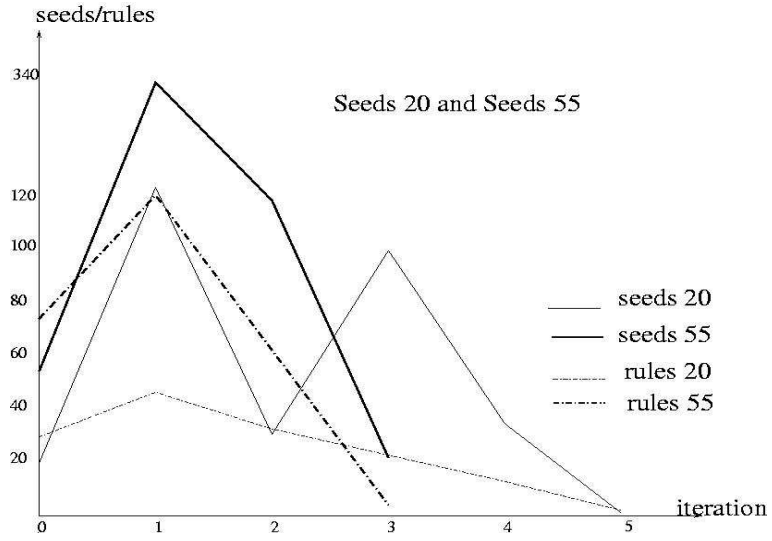

Figure 6.5: Iteration process of run 2 and 3 (20 and 55 seeds)

The choice of the management succession domain allows the comparison with other methods using the same corpus. Our result with 20 seeds (precision of 48.4% and recall of 34.2%) is comparable with the best result reported by Greenwood and Stevenson (2006) with the linked chain model (precision of 43.4% and recall of 26.5%). The linked chain model (Greenwood and Stevenson 2006) outperforms other automatic pattern learning systems, namely, the SVO

model (Yangarber 2001), the chain model (Sudo et al. 2001) and the subtree model (Greenwood and Stevenson 2006). However, a fair comparison is not possible. As already discussed in Chapter 5, our pattern representation can be used directly as the relation extraction rules. The pattern rules in other pattern learning systems (SVO, chain, linked chain or subtree model) can only serve as the trigger parts of the extraction rules. Furthermore, our result is more informative and precise than these systems: the relation instances are not only restricted to binary relations and furthermore all arguments are associated with their respective semantic roles.

6.5 Connectedness between Instances and Patterns

If we look closer at the seed-driven bootstrapping of pattern learning and instance extraction, the whole process can be described as a bipartite graph where the nodes are either instances or patterns, and connectivity between instances and patterns is detected by systems such as ***DARE***. Figure 6.9 (p. 134) illustrates a fraction of such a graph, where the error spreading is highlighted. We will discuss the error spreading issue in the next section.

To achieve good performance, the ***DARE*** system is expected to find seed instances leading to many patterns and patterns leading to many instances thus serving as hubs in the learning process, following the duality principle. It means that the hub instances or hub patterns build relevant nodes in the graph. The figures in Section 6.4 depict the iteration process of the ***DARE*** system for different system configurations. The interplay between the seeds and rules implies Zipf's law[5](see Figure 6.6), namely, some rules extract most instances, hence, the peak in the development curve of the instance discovery. However, the iteration processes do not explicitly reflect the connectedness between instances and patterns.

A further study was conducted to investigate the differences between the Nobel Prize domain and the management succession domain. For the Nobel Prize domain, we take the Nobel Prize A+B data set as our experimental example, because it delivers the best performance. Thus, we also select the data set in the management succession domain with the best performance, namely, the 55

[5] `http://en.wikipedia.org/wiki/Zipf's_law`

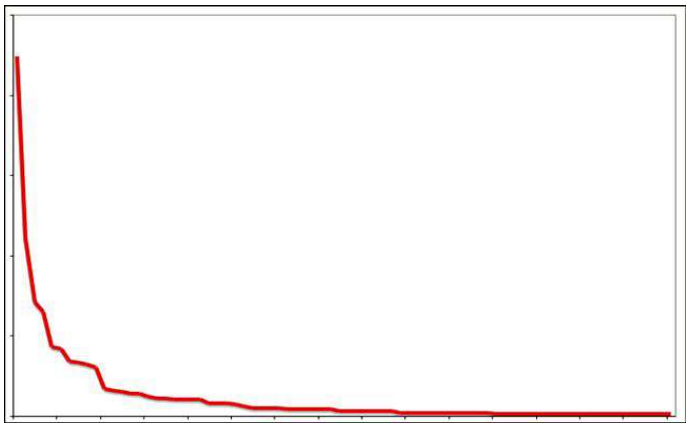

Figure 6.6: Zipf's law distribution

seed experiment. In the following figures, we show two features of the domain data:

- how many instances a pattern can extract (see Figure 6.7)
- how many patterns can be learned from an instance (see Figure 6.8)

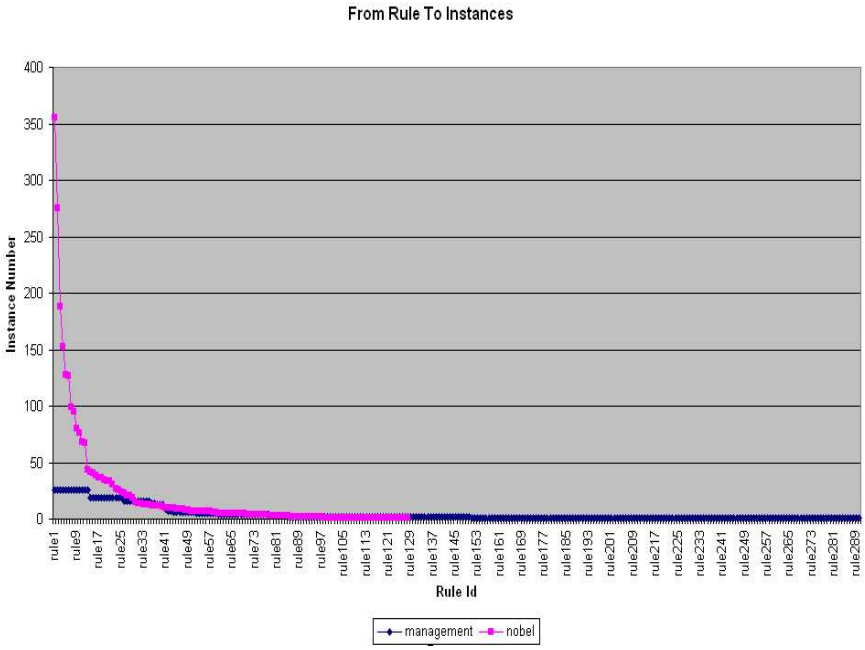

Figure 6.7: Distribution of instances extracted by patterns

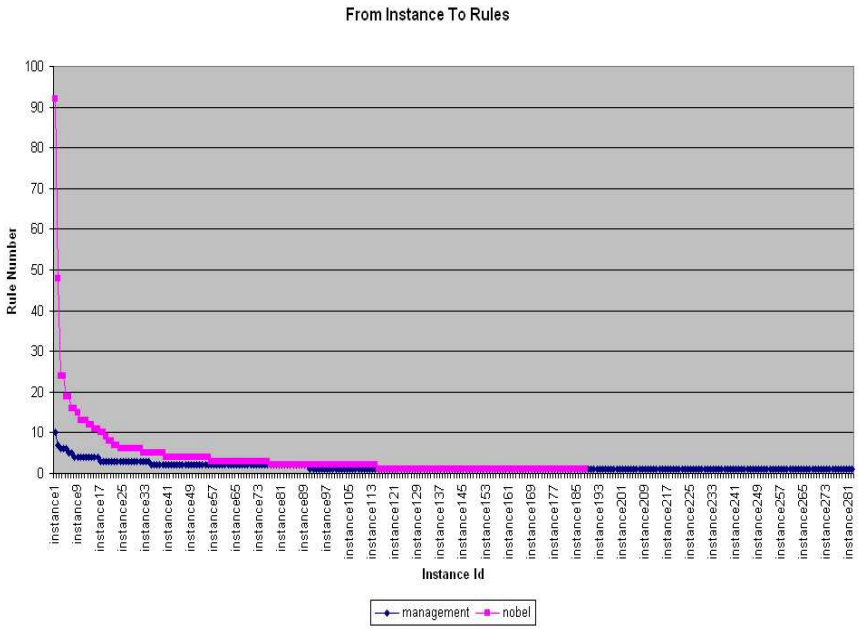

Figure 6.8: Distribution of patterns learned by instances

In comparison with the management succession domain, the skewed degree distribution can be shown for both patterns and instances in the Nobel Prize domain. Therefore most nodes in the graph can be reached in a few steps. Thus, even with a single instance as seed, the ***DARE*** system performs well in this domain.

The connectivity behavior in the management succession is completely different than the Nobel Prize domain. Its patterns and instances have a very small degree of connectivity. Thus, we need more instances as seed to discover enough patterns. It is clear that the distribution of mentions to events in the Nobel Prize domain data more closely follows the Zipf's law distribution than the data in the management succession domain. Therefore, our approach performs well with such a data property[6].

[6]In an invited talk based on our joint research work, Uszkoreit (2007) also reported on the insights discussed here.

6.6 Qualitative Analysis

We have shown that the Nobel Prize domain data, in particular, the largest corpus (see *Nobel Prize A+B*), possesses the most suitable data property for the ***DARE*** system, therefore obtaining the best performance. In this section, we will further investigate the learning process for this domain in order to better understand the mechanisms and effects of the ***DARE*** approach.

6.6.1 Detailed System Process Behavior

i.	seed	sentence	rule	new rule	induced rule	extracted instance	new seed	seed precision %	instances after merging
0	1	15	5	5	1	61	46	96.00	
1	46	330	77	75	39	439	130	91.50	
2	130	2759	398	353	162	663	112	89.00	
3	112	2440	392	200	100	121	57	84.21	
4	57	2009	233	33	23	107	25	100,00	
5	25	156	18	7	7	130	0		
total	371	7709	1123	673	332	1521	370		272

Table 6.11: Detailed system process behavior

Table 6.11 reports the system output after each iteration. In the first iteration, only one seed is applied, 15 relevant sentences are detected. Five pattern rules are derived from the 15 sentences. One rule is induced from these five new rules. This rule has extracted 61 new relation instances, from which 46 are selected as seeds for the next iteration after applying the filtering and ranking method. In the experiment, we allow only relation instances with three arguments as new seeds, to ensure the seed quality. Although the general trend of seed quality shows a decline, the precision values of new seeds are still very high. The total number of the learned rules is 1123 including the redundant ones. 332 rules have been induced and generalized from the 673 distinctive rules. 1521 relation instances have been extracted. After ranking and filtering, only 576 instances are returned. The template merging component unifies the compatible relation instances and delivers 272 relation instances as the final results.

Table 6.12 presents the distribution of relation instances with various complexities in the result set, which is compatible with the study of the seed complexity and performance reported in Table 6.3 (p. 110). The quaternary relation in-

stances exhibit the highest precision and recall value. Among the projections with three arguments, the combination of person, prize and area delivers the best performance as already reported in Table 6.3.

arity	correct	incorrect	precision %	recall %
1	1	0	100.0	0.3
2				
⟨person,prize⟩	48	21	69.5	13.7
3	84	25	77.0	23.9
⟨person,prize,area⟩	58	25	69.8	16.5
⟨Person,prize,year⟩	25	0	100.0	7.1
⟨Person,area,year⟩	1	0	100.0	0.3
4	87	6	93.5	24.8

Table 6.12: Distribution of relation complexity in the result set

An investigation was conducted to evaluate the quality of the learned pattern rules (see Table 6.13). We divide the pattern rules into four groups: *good*, *useless*, *dangerous* and *bad*. The good rules are rules that extract only correct instances, while bad ones produce exclusively wrong instances. Useless rules are those that do not detect any new instances. The dangerous rules are dangerous because they sometimes extract wrong instances. Most rules (83%) turn out to be useless. Most of these are too specific for the detection of new instances. The good rules make up 11.7%. Most of them extract three to four arguments. Only 1.6% are bad rules and 3.7% dangerous.

6.6.2 Sentence vs. Paragraph

In the current system experiment, we have not attempted any discourse analysis. All event instances are extracted from sentences. The total number of instances that can be extracted from the sentences is 350 Nobel Prize winner events. Our evaluation has taken these 350 instances as the gold-standard value for the Ideal table. However, as discussed in Chapter 5, arguments belonging to a relation instance are often distributed over several sentences. These sentences are usually linked by coreferences, semantic chains or various discourse relations. If we also consider relation instances expressed via various sentences, the total corpus mentions 392 relation instances. These distributed instances are nevertheless contained in a paragraph such as in (6.4):

arity	useless	bad	dangerous	good
4	105	3	2	18
3				
⟨person, prize, year⟩	31	0	1	12
⟨prize, year, area⟩	8	0	2	1
⟨person, prize, area⟩	307	6	7	35
⟨person, year, area⟩	11	0	0	0
2				
⟨prize, year ⟩	5	0	0	2
⟨person, prize⟩	43	2	7	6
⟨person, year⟩	8	0	0	2
⟨year, area⟩	4	0	1	0
⟨prize, area⟩	10	0	2	0
⟨person, area⟩	26	0	3	3
sum	558	11	25	79
relative to total rules	83%	1.6%	3.7%	11.7%

Table 6.13: Evaluation of rule quality and their distribution

(6.4) 1) Three of the ***Nobel Prizes*** for ***Chemistry*** during the first decade were awarded for pioneering work in organic chemistry.
2) In ***1902 Emil Fischer*** (1852-1919), then in Berlin, was given the prize for his work on sugar and purine syntheses.
3) Fischer's work is an example of the growing interest among organic chemists in biologically important substances, thus laying the foundation for the development of biochemistry, and at the time of the award Fischer mainly devoted himself to the study of proteins.
4) Another major influence from organic chemistry was the development of the chemical industry, and a chief contributor here was Fischer's teacher, ***Adolf von Baeyer*** (1835-1917) in Munich, who was awarded the prize in ***1905***.

In example (6.4), two concrete Nobel Prize winning event instances in Chemistry are mentioned, one in the year 1902 for Emil Fischer and another in 1905 for Adolf von Baeyer. However, the linking between the Nobel Prize winners with the Nobel Prize is expressed indirectly via the anaphoric expression *the prize*. The two arguments (*prize name* and *area*) shared by the two event instances are located in the first sentence. The two winners and their prize award years can be found in sentence two and four, respectively. If we consider sen-

tence two and four independently from the context, we cannot tell that they are about the Nobel Prize events, without resolving the anaphoric reference *the prize* as the Nobel Prize.

6.6.3 Error Analysis

We also performed a systematic analysis of incorrectly extracted relation instances[7]. Error reasons can be classified in four groups:

- ***content***: Wrong facts are expressed by the corpus sentences
- ***modality***: The facts or events are embedded in a scope of a modality which either denies or weakens the truth value of the facts or events, e.g, negation or wish.
- ***NLP annotations***: the NLP components deliver a wrong analysis or cannot analyse the sentence.
- ***rule***: the learned rules lead to wrong seeds

content %	modality %	*SProUT* %	MINIPAR %	*SProUT* & MINIPAR %	rule %
11.8	17.6	5.9	38.2	11.8	14.7

Table 6.14: Distribution of error types

Table 6.14 reports the distribution of the error types. More than half of errors (55.9%) are caused by the wrong NLP analysis. The biggest error source is the parsing system MINIPAR, namely, 38.2% are because of the wrong dependency structures. 5.9% errors are made by the named entity recognition system *SProUT*. The interface between *SProUT* and MINIPAR has generated 11.8% errors.

Sometimes, a newspaper article reports a Nobel Prize winner event with wrong areas or wrong award years. In the following, we give two examples:

(6.5) 1. **wrong area**: But the society's position drew a stinging rebuke from Dr. Paul Berg, who won the Nobel Prize in *Medicine* in 1980. (*Chemistry* is the correct area)

[7] Li (2006) reported some of our initial error analyses.

2. **wrong year**: The Dalai Lama, who won the Nobel Peace Prize in *1985*, heads a government in exile based at the northern Indian town of Dharamsala, where more than 100,000 Tibetan refugees now live.
   (*1989* is the right year)

Errors caused by the wrong data can be detected by the Ideal table evaluation, because the Ideal table contains correct facts or events independent of the input texts. However, IE systems should be able to extract wrong facts or events, if the input texts report them. The validation of the truth value of the extracted facts or events is beyond the standard IE tasks.

Modality is an important aspect for high precision IE. In the current experiment, we did not develop special methods of dealing with the modality problem. Therefore, the extracted results are not valid when they occur in the scope of modalities that do not support the truth values of the mentioned facts or events. The following examples show modalities expressed in a variety of ways, e.g., by a noun such as *speculation*, or by modal adjective or adverb such as *possible* or *never*, or even by some fictive contexts provided by films or novels. The linguistic structures embedded in the modality scopes are highlighted with brackets. The second sentence poses an additional challenge because of irony. Sentence four introduces a fictive Nobel Prize winner, *Josiah Bartlett*, broadcasted by a TV program. Thus, world knowledge is needed here to resolve the modality.

(6.6)
1. The talk has included *speculation*[that North Korean leader Kim Jong Il and South Korean President Kim Dae-jung might win the Nobel Peace Prize for their step toward reconciliation, the most promising sign of rapprochement since the Korean war ended with a fragile truce in 1953].
2. It's also *possible* [that O.J. Simpson will find the real killer, that Bill Clinton will enter a monastery and that Rudolph Giuliani will win the Nobel Peace Prize].
3. Detractors have long pointed out, for example, [that Freud *never* won the Nobel for medicine], and that [Chekhov, Proust and Conrad are among the giants who *never* won for literature].
4. In NBC's "West Wing," [we get President Josiah Bartlett, a Nobel Prize-winning economist who is a faithful husband, fabulous dad

and forgiving boss].

As mentioned above, the weakest component in the ***DARE*** system is the NLP analysis, in particular, the dependency analysis, although MINIPAR belongs to reliable analyzers among the new class of relatively deep robust parsers. Sometimes MINIPAR establishes wrong links between linguistic structures. For example (6.7), there are three parallel appositional noun phrases about three persons *William Crowe*, *Hans Bethe* and *Herbert York*. The last two noun phrases are connected via the conjunction "and". The apposition of *Hans Bethe* describes him as a Nobel Prize winner. MINIPAR is overeager in this case and links the apposition of the second name with the third name using "and" as their connector. This breaks the relationships between the second name and its opposition.

(6.7) William Crowe, former chairman of the joint chiefs of staff; Hans Bethe, [*the Nobel Prize-winning physicist, and Herbert York*], a former founding director of the Livermore National Laboratories sent letters to the Senate urging action on the treaty now.

A similar problem occurs in the sentence below (see example (6.8)): the closest simple noun phrases around the conjunction are connected with each other at first, thus yielding a wrong dependency structure. This parsing strategy is not suitable for newspaper texts where quite often complex noun phrases are coordinated by a conjunction.

(6.8) In a recent paper World Bank President [*James Wolfensohn and Nobel Prize economist*] Amartya Sen sketched the plight of the bottom half of the globe's peoples: "Three billion people live on less than two dollars a day, 1.3 billion (one human out of four) do not have clean water, 130 million children do not go to school, and 40,000 children die every day because of hunger-related diseases.

There are also error cases caused by the interaction between errors generated by *SProUT* and MINIPAR. Example (6.9) contains a Nobel Prize winning event, namely, Dr. E. Donnall Thomas obtaining the Nobel Prize in medicine in 1990. However, our system recognizes *Fred Hutchinson* as the winner. *SProUT*

recognizes *Fred Hutchinson* as a person name instead of as a location or an organization, and MINIPAR combines the relative pronoun *who* with this wrong person name. The subject of the verb "win" is then resolved as *Fred Hutchinson* by the parser.

(6.9) "I haven't seen the data yet, but we've been told they basically found no significant difference between transplantation and routine chemotherapy," said Dr. E. Donnall Thomas, the former clinical director at [*Fred Hutchinson who won the 1990 Nobel Prize in medicine for pioneering the bone marrow transplant*].

We are relieved to see that only 14.7% of the errors come from wrong rules. Most of these errors are generated by rules headed by the verb "nominate". In Section 6.3, we discussed the consequence of ambiguous seeds. If a seed is ambiguous, it also triggers rules that learn other relations. In the Nobel Prize domain, we are faced with the problem that it is common sense that all Nobel laureates are nominated before they won the prizes, but not all nominated persons are Nobel laureates. Given a seed, we cannot avoid learning rules that also mention nomination events. In the following subsection, we discuss the error spreading degree of wrong rules.

6.6.4 Error Spreading during Bootstrapping

For the bootstrapping process, we checked step by step where incorrect patterns or seeds are hypothesized and furthermore, whether these wrong information sources proliferate.

Figure 6.9 (p. 134) depicts the error spreading within one entire learning and extraction process. The red colored picture elements are the error spreading areas, either bad rules or incorrect seeds or incorrect found instances. The black colored rules are useless ones. The orange rules are dangerous rules that produce both correct and wrong instances. The blue elements are correct rules and instances.

It turned out that 94% of the incorrect seeds produce no further patterns, thus, no dangerous relation instances occur because of them. The only problematic rule originating from a wrong seed is the rule headed by the verb "nominate".

(6.10) [$_{rule}$ "nominate": ⟨**object**: recipient⟩, ⟨**mod**: prize, area ⟩]

This rule has given rise to three additional incorrect instances. Like the other wrong seeds, these three do not generate new pattern rules in the next iteration.

However, correct seeds can also produce pattern rules that extract incorrect seeds or correct seeds that lead to further dangerous or bad rules. As listed in Table 6.13, among the set of pattern rules, only 36 rules (5.3%) generate incorrect instances. Most of them, namely, 31, are derived from correct seeds. 23 rules often extract incorrect seeds in addition to correct ones, while eight exclusively detect incorrect seeds. The longest life cycle of these wrong pattern rules is three iterations. Most incorrect seeds are generated by rules such as example (6.11). When (6.11) applies to examples (6.7) and (6.8), wrong relation instances are produced. In this case the wrong pattern rules match the wrong dependency structures.

(6.11) [$_{rule}$ "and": ⟨**person**: recipient⟩, ⟨**NP**: prize, area⟩]

Since the majority of incorrect patterns fortunately do not give rise to further instance detection, we could concentrate on a few cases that indeed lead to the proliferation of incorrect results. We expect that we will be able to modify the rule extraction algorithm in such a way that many of these cases can be avoided.

Our system delivers 83% useless rules. It turns out that in a number of cases, adverbs, adjectives, noun phrases or prepositional phrases that do not belong to the appropriate relation detection pattern are included in the pattern hypothesis. These rules are too specific to apply to new data. Additional tree generalization methods such as node pruning or node clustering is needed to make the rules more general, thus more useful.

6.7 Extensions

6.7.1 Nobel Prize Domain as a Carrier or Bridge Domain

As mentioned above, the Nobel Prize is one of the most prominent prizes with extensive media coverage leading to the desired high degree of redundancy in mentions. Patterns learned for the Nobel Prize should be generic enough to extract relations for other prizes and awards too. Indeed these patterns turn out to be especially helpful to detect less prominent and less mentioned prizes and awards. We construct three scenarios to see whether the learned patterns are applicable for extraction of additional prize winning events and similar relations. In the first scenario, we apply the patterns to the same corpus to acquire other prize winning events. In the second scenario, we remove the entity restriction of the "prize name" in the corresponding pattern slots and allow the prize name slot to be filled with any noun phrases, even if they are not recognized as prize names. The motivation is to detect prizes and awards that are not discovered by the entity recognition system. In the third scenario, we apply the learned patterns to a domain corpus on music and musicians with the aim of extracting music award events and to learn new pattern rules. This experiment has been carried out as part of a bachelor thesis (Felger 2007) supervised by the author.

In the first scenario, a list of Prize winning events has been extracted. The most frequently detected prize is the Pulitzer Prize. We have detected 97 Pulitzer Prize winning event instances. Among them 95 are correct. Similarly to the Nobel Prize, the prize winners obtain the Pulitzer Prize for some special area in literature, e.g., poetry. The precision of the Pulitzer Prize detection is 97%. We find also the winning events about the following prizes that are recognized by *SProUT*:

- albert lasker award
- pritzker prize
- turner prize
- prix_de_rome

The event instances of the above prizes are mentioned very seldomly in the corpus. Only one to three instances for each prize were found.

Prize and Award	Other
Academy Award	$ 1 million
Cannes Film Festival's Best Actor award	about $ 226,000
American Library Association Caldecott Award	acclaim
American Society	discovery
Blitzker	doctorate
Emmy	election
feature photography award	game
the first Caldecott Medal	master's degree
Francesca Primus Prize	presidency reelection
gold (gold metal)	scholarship
National Book Award	.
Oscar	.
P.G.A	.
PEN/Faulkner Award	.
prize	
reporting (the investigative reporting award)	
Tony (Tony Award)	
U.S. Open	

Table 6.15: Second scenario: fuzzy extraction

In the second scenario (which we call *fuzzy extraction*), we find more awards, even less well-known ones, and also other wins, e.g., money and praise, as shown in Table 6.15. The precision of our extraction task here is 73%.

In the third scenario, we first conduct a survey of web sites in order to find useful web sites for the relevant relations in the musician domain. We select the top 100 and the bottom 100 musicians available in a music database provided by Research Studios Austria (ARC). We combine the musician names (NAME) with some relevant keywords such as "NAME news", "NAME music news", "NAME award", "NAME prize", "NAME winner". It turns out that the top musicians are more frequently distributed in some general public websites such as wikipedia. The bottom musicians are mentioned more often in blogs such as `myspace.com`. This result can be potentially taken into account when it comes to detecting rising stars. This musician corpus is in comparison to the Nobel prize domain corpus less redundant. An initial evaluation was carried out to compare the system performance with and without the Nobel Prize rules. It turns out that two thirds of the total instances are discovered by the rules learned in the Nobel Prize domain.

All three scenarios confirm the carrier function of a more fertile sibling domain. The patterns learned by the Nobel Prize domain are generic enough to be applicable to other awards. In particular, a prominent sister domain helps to extract more instances than could be extracted by learning from the actual target domain.

6.7.2 Domain Independent Binary Relations

The additional positive side-effect of the ***DARE*** system is that it also learns rules for binary relations. Most of these are domain independent and can be reused for other domains. For example, in the management succession domain, the binary relations such as persons and their positions, and persons and their affiliations (organizations) are domain independent. The evaluation of the binary relation extraction delivers 98% precision value. As mentioned above, there are no binary relations between positions and organizations.

6.8 Conclusion

Several parameters are relevant for the success of a seed-based bootstrapping approach to relation extraction. One of these is the arity of the relation. Another one is the locality of the relation instance in an average mention. A third one is the type of the relation arguments: Are they named entities in the classical sense? Are they lexically marked? Are there several arguments of the same type? Both tasks we explored involved extracting quaternary relations. The Nobel Prize domain shows better lexical marking because of the prize name. The management succession domain has two slots of the same NE type, i.e., persons. These differences are relevant for any relation extraction approach.

The success of the bootstrapping approach crucially depends on the nature of the training data base. One of the most relevant properties of this data base is the ratio of documents to relation instances. Several independent reports of an instance usually yield a higher number of patterns. The two tasks we used to investigate our method differ drastically in this respect. The Nobel Prize domain was selected as a learning domain for general award events since it exhibits a high degree of redundancy in reporting. A Nobel Prize triggers more news reports than most other prizes. The results achieved met our expectations.

With one randomly selected seed, we could finally extract most relevant events in some covered time interval. However, it turns out that it is not just the average number of reports per event that matters but also the distribution of reportings to events. Since the Nobel Prize data exhibits a certain type of skewed distribution, the graph exhibits properties of scale-free graphs. The distances between events are shortened to a few steps. Therefore, we can reach most events in a few iterations. The situation is different for the management succession task where the reports came from a single newspaper. The ratio of events to reports is close to one. This lack of informational redundancy requires a higher number of seeds. When we started the bootstrapping with a single event, the results were rather poor. Going up to twenty seeds, we still did not get the performance we obtained in the Nobel Prize task but our results compare favorably to the performance of existing bootstrapping methods.

The conclusion we draw from the difference observed between the two tasks is simple: We shall always try to find a highly redundant training data set. If at all possible, the training data should exhibit a skewed distribution of reports to events. Actually, such training data may be the only realistic chance for reaching a large number of rare patterns.

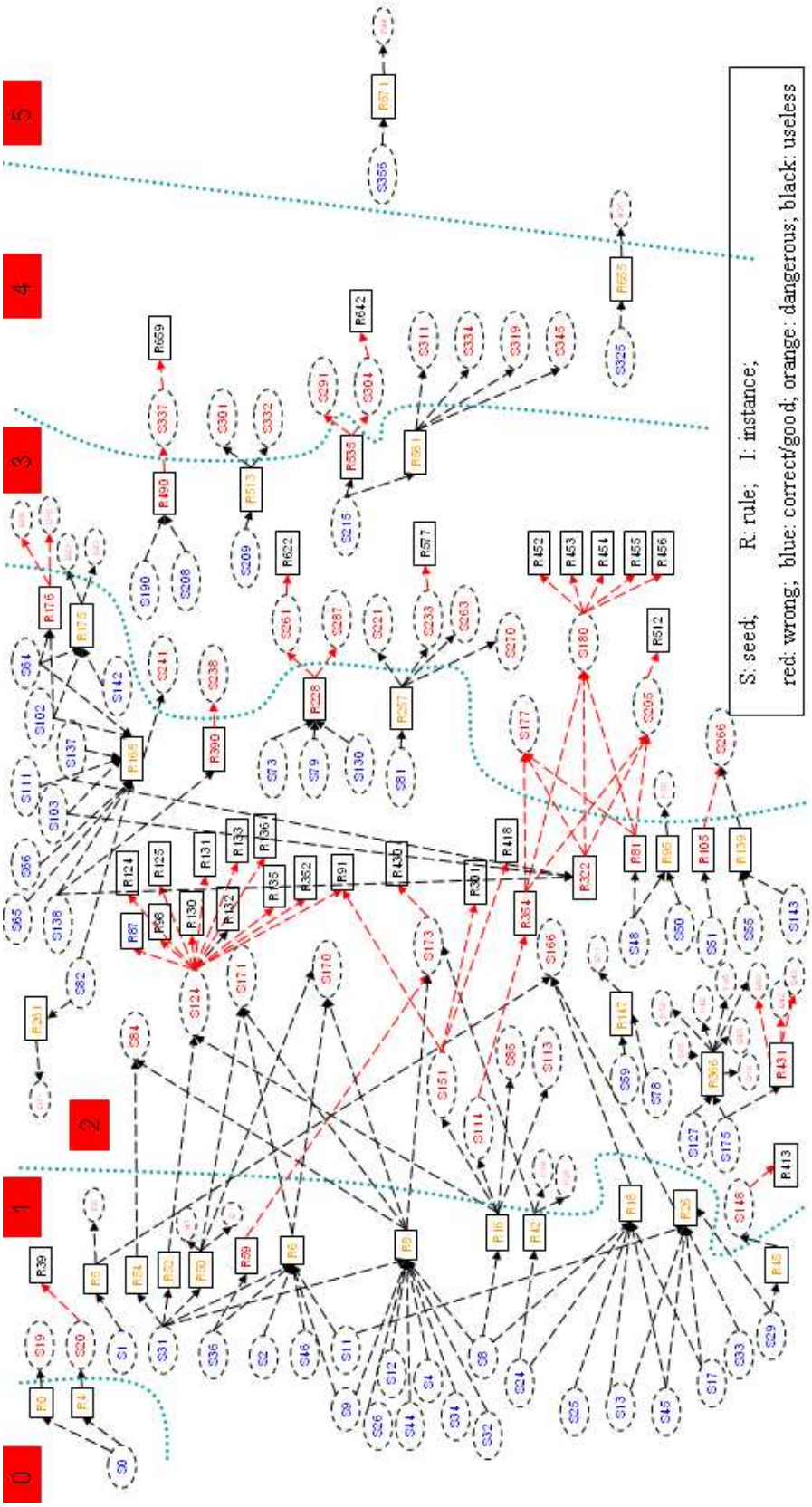

Figure 6.9: Error spreading during learning and extraction

Chapter 7

Conclusion and Future Work

This thesis has designed and implemented a relevant task of Information Extraction (IE), i.e., the extraction of relations from large volumes of natural language texts. IE can be regarded as a pragmatic approach to text understanding. Its semantic models describe the application requirements that can always be viewed as populating databases with detected entities and relations detected among them. The ***DARE*** system automatically learns and discovers relation extraction grammar rules, utilizing a limited set of target relation instances as initial knowledge. These relation extraction grammar rules map linguistic expressions, or actual linguistic structures delivered by the NLP systems, to the target semantic relations. The learning algorithm is driven by the target semantic structure and enables inexpensive adaptation to new relation extraction tasks and domains. Our solutions contribute to the perspective that restricted textual understanding can be realized as a bottom-up extraction of relations with various complexities. From IE point of view, the final result of text understanding can be simplified as a set of relations or events, which are connected via ontological structures. In this chapter, we will summarize the major features of the ***DARE*** framework and emphasize its scientific contributions to the IE research area including relevant new insights. Furthermore, we will discuss the open issues and propose some future directions.

7.1 Summary

The thesis describes a further development of Xu et al. (2007) both at the theoretical and the implementation level. It describes a minimally supervised machine learning framework for extracting relations of various complexity. The system starts from a small set of n-ary relation instances as "seeds" in order to automatically learn pattern rules from parsed data which can then extract new instances of the targeted n-ary relation and its projections. We propose a novel rule representation model which enables the composition of n-ary relation rules on top of the rules for projections of the relation. The compositional approach to rule construction is supported by the bottom-up pattern extraction method. Because we only consider linguistic structures that contain arguments in the seed relations, the pattern extraction does not suffer from the computational problem caused by the large rule productivity, which is a weakness of other rule representation models. In comparison to other automatic approaches, our rules can not only localize relation arguments but also assign their exact target argument roles. The learning step is embedded in a bootstrapping process in which the number of rules and seed relations increases iteratively. Systematic evaluations are conducted to assess general system performance such as precision and recall, and also its detailed system development behaviors during the bootstrapping process. The comparison of two different application domains (Nobel Prize awards and management succession) demonstrate that the properties of the data play an important role for the feasibility and performance of the ***DARE*** framework. The ***DARE*** system delivers very promising results for the Nobel Prize corpus and still compares favorably with existing systems in the much more difficult domain of management succession reports.

7.1.1 Semantic Seed

The only domain knowledge of the ***DARE*** framework is the semantic seed that embodies the target semantic relation and its complexity. The decision to use relation instances as seed instead of linguistic patterns allows the ***DARE*** framework to be flexible with respect to the NLP components to be employed. These may be shallow, deep or hybrid processing components. Furthermore, a semantic seed helps localize the textual fragments where linguistic patterns are potentially embedded. A semantic seed is a more natural anchor for detection

of the adequate textual units for a relation instance, since the arguments of a relation instance are often distributed in more than one sentence. Above all, the explicit semantic role information of the arguments in the seed relation can be utilized to annotate the linguistic arguments in the learned pattern rules. Thus, the pattern rules in the ***DARE*** system can be qualified straightforwardly as extraction rules.

The semantic specificity of a seed plays an important role in the system performance. An underspecified relation instance as seed is often ambiguous and gives rise to learning extraction grammars that potentially refer to other relations, while an overly specific relation instance with some additional optional arguments either cannot match texts or triggers rules that are too specific or too complex and therefore can not be applied to new texts. Therefore, we propose to choose the smallest number of arguments which taken together most probably express the relation. Further, we also suggest selecting relation instances as seeds that represent the target relation unambiguously, if possible. However, there are some domains where the same argument tuple can present different relations, or the same argument instance plays different roles. For example, all Nobel Prize winners had been also involved in nomination events preceding winning the award, and a person often takes over a new position after leaving an old position, thus often filling two different roles at the same time. Therefore, redundancy is one of the requirements of the information sources, helping crystallize patterns for the right relation type.

The brief study in Section 6.3 shows that the more arguments a sentence contains, the more probably the sentence refers to the seed event: a good indication for relevant sentence retrieval. However, most relevant sentences contain only a subset of the arguments, referring to projections of the events. Thus, the projections have to be taken into account to improve the recall value.

7.1.2 Rule Representation

One of the major contributions of the ***DARE*** framework is the ***DARE*** pattern rule representation. This rule representation is compositional and can be recursive, thus, complex rules are made of simpler rules which themselves can contain simpler rules. Such a general rule representation permits adaptation to any relation types with any complexity. In practice, it is not bound to any spe-

cific linguistic representation unlike other pattern representation models, e.g., SVO model, the chain, the linked chain and the subtree models discussed in Stevenson and Greenwood (2006).

The compositionality feature and its independence with respect to a concrete linguistic analysis make the ***DARE*** rule representation very powerful and much more expressive than all other models. It is able to cover all linguistic constructions that express the semantic relations and can also assign the semantic role information to the linguistic arguments that serve as the slot fillers. From the theoretic point of view, the ***DARE*** rule representation can reach the full coverage directly or indirectly, if the data property of a corpus is a graph where all nodes (instances and patterns) are connected with each other, thus not containing any isolated subgraphs.

Since the pattern discovery process in ***DARE*** is driven by the target semantic relation, the rule productivity does not behave exhaustively, like the ExDisco system for the SVO construction (Yangarber 2001), and the systems for the chain, the linked chain and the subtree model (Stevenson and Greenwood 2006). These other systems extract all linguistic constructions obeying their pattern representations dominated by verbs. All of them produce a much larger number of rules, in particular, the chain, the linked chain and the subtree models. The subtree model is an extreme approach with a huge number of pattern rules. For the largest corpus in the Nobel Prize domain, we only produce 1123 rules, for the MUC-6 corpus 263 rules with 55 seeds, which are only a fraction of the rules discovered by other systems for the same management succession corpus (see Table 3.2 on page 43). The ***DARE*** system evaluation reported in Section 6.4 shows that the ***DARE*** system achieves comparably promising performance with a much lower rule productivity.

Thus, the ***DARE*** representation has on the one hand fulfilled the coverage and expressiveness requirement and on the other hand avoided the production of large numbers of useless patterns.

7.1.3 Pattern Extraction

A further contribution of the ***DARE*** framework is the bottom-up rule extraction algorithm which supports the ***DARE*** rule representation. This algorithm

extracts complex rules for the target relation on top of the rules for their projections in a compositional way. The result of the ***DARE*** pattern extraction component is a set of rules which extract the target relations and their projections.

The evaluation analysis in Section 6.6 proves that rules for projections are relevant for system performance, in particular, the recall value. For the largest Nobel Prize corpus, 60% of the extracted relation instances are projections of the target relation (see Table 6.12, p. 123). 77% of the good pattern rules are rules for relation projections (see Table 6.13, p. 124). Therefore, a successful relation extraction system for complex relations such as events has to include the functionality of discovering projections as well. This move may help to overcome the frustrating performance barrier that relation extraction has been faced with for many years now.

Another side effect of learning projection rules is to discover domain independent and reusable rules such as binary rules for person-position, person-affiliation, as seen in the management succession task (see Section 6.7).

7.1.4 Rule Induction and Generalization

Another novelty of the ***DARE*** system is the bottom-up rule induction and generalization method that first constructs the simpler rules and then combines them into more complex ones. Based on this strategy, the ***DARE*** rule learning method sets up a general framework for rule induction and generalization. This opens up new options in the area of IE rule learning, because many simple rules for projections can be shared among IE rules for different relations. In this way such component rules can be learned or reused even if the corpus does not contain any mentions from which the composed rules could have been learned for the actual target relation. Therefore this approach may offer a solution to some types of data sparseness.

7.1.5 Data Property

Although the ***DARE*** rule representation is very expressive and can ideally cover all linguistic constructions that can be utilized as pattern rules, the dis-

crepancy in the system performance between the Nobel Prize award domain and the management succession domain points out that the ***DARE*** system, or more generally, the bootstrapping framework, is more suitable for some types of data than for others. Even within the Nobel Prize corpus, the performance improves when the data size increases. Uszkoreit (2007) raised important questions with respect to the influence of the data property on the feasibility of the bootstrapping method for the ***DARE*** system:

1. Why does it work for some tasks?
2. Why doesn't it work for all tasks?
3. How can we estimate the suitability of domains and data?
4. How can we deal with less suitable domains or data sets?

The analysis of the connectedness between patterns and instances for the two experiment domains (see Section 6.5) provides answers to some of the above questions. The connectedness between the patterns and the instances and the connectivity degree of a single pattern or an instance to other patterns or instances in the Nobel Prize data corresponds to a skewed long-tailed distribution. Thus, few patterns and few instances in the Nobel Prize domain data exhibit a high degree of connectivity and can therefore serve as hubs to most other nodes in the graph. Even with one instance as seed, the ***DARE*** system performs well in the Nobel Prize domain. This data property is expected by the ***DARE*** system design according to the duality principle, namely, the ***DARE*** system should find seed instances which can find many patterns or patterns by which many instances are expressed. The connectivity behavior in management succession is very different from the Nobel Prize domain. Most patterns as well as most instances show a very low degree of connectivity. Thus, we need more instances as seeds to discover enough patterns. In this case the data properties do not support the duality principle.

The empirical results gained in our evaluation confirm the research results reported by Jones (2005). Given a corpus with a small world data property (Amaral et al. 2005), all nodes in the graph are theoretically reachable in few steps if the discovery mechanism is powerful enough, like our ***DARE*** rule representation model. If the node degree of the set of initial and detected seeds follows a skewed distribution, the probability of finding most nodes is very high.

Learning corpora possessing this data property constitute ideal scenarios for the application of the ***DARE*** system. In such a scenario, the cardinality of the initial seed does not matter much for the overall system performance.

7.2 Next Steps and Future Work

Although this thesis reports on a completed system and a completed evaluation, the findings have inspired a host of new ideas for follow-up research. The new ideas and planned future research steps can be divided into three groups: improvement of the recall value, boosting the precision value, and further potential applications of the ***DARE*** framework.

7.2.1 Improvement of Recall

7.2.1.1 Data Property

As discussed above, the data property is a very relevant factor for ***DARE*** system performance. The management succession domain has a relatively low recall suffering from poor redundancy: nearly all events are just mentioned once, since the data is from a single newspaper, namely, the New York Times. In Xu and Uszkoreit (2007) and Uszkoreit (2007), several strategies have been proposed to circumvent the bad data property problem.

A general and direct approach is to utilize the web to increase redundancy, as also independently proposed by Blohm and Cimiano (2007).

Another strategy is to enlarge the domain or utilize some prominent sibling domains as carrier domains. This requires the modelling of relevant ontological relationships between different domains. For example, the Pulitzer Prize award domain belongs to the Prize award domain, having the Nobel Prize award as its prominent sibling domain. The experiments reported in Section 6.7 show that the Nobel Prize patterns are general enough to help discover Pulitzer Prizes and prizes for musicians.

A further option is to make use of the compositional property of the ***DARE*** rule representation. The target relation can be broken down into a group of

projections. The ***DARE*** system can learn projection rules that are available in other domains with suitable and better data properties. An additional rule generation component can be developed to construct relation rules on top of the projection rules.

7.2.1.2 Rule Generalization

Table 6.13 (p. 124) reported that 83% of the learned pattern rules are useless. Most of them are too specific to apply to new texts. This means that there is a great potential for improving the rule induction and generalization method. We plan to apply generalization methods at various levels, such as lexical as well as syntactic.

7.2.1.3 Discourse Analysis

A great research challenge is the integration of discourse analysis into the ***DARE*** framework. In the current system setup, only relation instances at the sentence level have been considered. A potential solution is to learn discourse level ***DARE*** rules from general discourse analysis results.

7.2.2 Boosting Precision

The analysis in Section 6.6 has identified four error sources for bad instances: wrong content, modality denying and weakening of truth value, wrong NLP analysis and bad rules.

A scientifically exciting topic is the learning of negative rules from negative examples. We assume that there will be two groups of negative rules: domain independent and domain specific. Negative rules describing the modality scopes can be domain independent and reusable for all relation extraction tasks. The domain specific rules will include rules detecting wrong relations, for example, the rules headed by the verb "nominate" in the Nobel Prize award domain. This experiment can reduce errors caused by wrong modalities and bad rules.

In our experiment, most errors stemmed from an incorrect NLP analysis. In initial experiments we have already started to extend our NLP analysis with some

high-precision deep NLP systems. We plan to extract patterns from RMRS with extended ERG (Copestake and Flickinger (2000), Copestake (2003), Zhang and Kordoni (2006) and Zhang et al. (2007)). Our first experiment yields relatively promising results, namely, 80% coverage for the Nobel Prize domain sentences and 61% for the management succession sentences[1]. It is important for us to study the overlap between the coverage of ERG and that of other relatively deep dependency parsers, and to assess the degree of the quality improvement provided by ERG. The robust dependency parsers can serve as baseline systems for dealing with sentences not covered by ERG. Furthermore, we will investigate

- the complexity of semantic relations in comparison to the depth of the general semantic representations,
- the influence of the local and non-local linguistic relations on the pattern rules and their projections,
- the discovery and development of mapping strategies between linguistic and semantic, in the sense of ontological, relations, with special focus on cases of ambiguity and underspecification.

7.2.3 Potential Applications

The experiments with two different domains have helped us gain valuable insights into the potential and the limitations of the ***DARE*** framework. In future research, we plan to apply ***DARE*** to more domains and even more complex tasks such as opinion mining or sentiment analysis. Therefore, the integration of discourse analysis and modality aspects will be necessary steps to prepare for these future applications.

We believe that the potential of our bootstrapping method for further application domains is large. We will conduct additional case studies and careful analysis of their respective performance, in order to arrive at convincing criteria that enable us to predict which combination of methods would be most useful for which tasks.

[1]The experiment is conducted by Yi Zhang, a colleague in the Computational Linguistics department at Saarland University

Bibliography

Abney, S. (2002). Bootstrapping. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 360–367.

Aduna B.V. (2004). *User Guide for Sesame.*

Agichtein, E., E. Eskin, and L. Gravano (2000). Combining strategies for extracting relations from text collections. In *Proceedings of the 2000 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2000).*

Agichtein, E. and L. Gravano (2000, June). Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries (DL'00)*, San Antonio, TX.

Agichtein, E., P. Ipeirotis, and L. Gravano (2003). Modeling query-based access to text databases. *Proceedings of the Sixth International Workshop on the Web and Databases, WebDB*, 87–92.

Agichtein, E., S. Lawrence, and L. Gravano (2001). Learning search engine specific query transformations for question answering. In *World Wide Web*, pp. 169–178.

Amaral, L., A. Scala, M. Barthélémy, and H. Stanley (2005). Classes of small-world networks. *Proceedings of the National Academy of Sciences 102*(30), 10421–10426.

Androutsopoulos, I. and G. Ritchie (2000). Database interfaces. In R. Dale, H. Moisl, and H. Somers (Eds.), *Handbook of Natural Language Processing.*

Appelt, D. (2003). Semantics and information extraction. Center for Language and Speech Processing.

Appelt, D. and D. Israel (1999). Introduction to information extraction technology.

Baader, F., D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider (2003). *The Description Logic Handbook*. Cambridge University Press.

Baker, C. F., C. J. Fillmore, and J. B. Lowe (1998). The Berkeley FrameNet project. In *Proc. of COLING-ACL*, Montréal, Canada.

Baldwin, T., E. Bender, D. Flickinger, A. Kim, and S. Oepen (2004). Road-testing the English Resource Grammar over the British National Corpus. In *Proc. of LREC*, Lisbon, Portugal.

Bechhofer, S., F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein (2004). OWL web ontology language reference. Technical report, W3C. 10 February.

Bikel, D., R. Schwartz, and R. Weischedel (1999). An Algorithm that Learns What's in a Name. *Machine Learning 34* (1), 211–231.

Blohm, S. and P. Cimiano (2007, September). Using the Web to Reduce Data Sparseness in Pattern-based Information Extraction. *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*.

Brickley, D. and R. V. Guha (2004). RDF vocabulary description language 1.0: RDF Schema. Technical report, W3C.

Brin, S. (1998). Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*.

Broekstra, J., A. Kampman, and F. van Harmelen (2002). Sesame: A generic architecture for storing and querying RDF and RDF schema. In *Proceedings ISWC 2001*, pp. 54–68. Springer.

Bunescu, R. C. and R. Mooney (2005, October). A shortest path dependency kernel for relation extraction. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, B.C., pp. 724–731.

Burchardt, A., K. Erk, and A. Frank (2005). A WordNet Detour to FrameNet. In *Proceedings of the 2nd GermaNet Workshop*.

Burchardt, A., A. Frank, and M. Pinkal (2005). Building Text Meaning Representations from Contextually Related Frames – A Case Study. In *Proceedings of the Sixth International Workshop on Computational Semantics, IWCS-06*, Tilburg, The Netherlands.

Califf, M. (1998). *Relational Learning Techniques for Natural Language Information Extraction*. Ph. D. thesis, PhD thesis, Tech. Rept. AI98-276, Artificial Intelligence Laboratory, The University of Texas at Austin, 1998.

Califf, M. and R. Mooney (1998). Relational learning of pattern-match rules for information extraction. *Working Notes of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, 6–11.

Califf, M. and R. Mooney (2004). Bottom-Up Relational Learning of Pattern Matching Rules for Information Extraction. *Journal of Machine Learning Research 4*(2), 177–210.

Califf, M. E. and R. J. Mooney (1999). Relational learning of pattern-match rules for information extraction. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, Orlando, FL, pp. 328–334.

Callmeier, U. (2000). PET–a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering 6*(01), 99–107.

Callmeier, U., A. Eisele, U. Schäfer, and M. Siegel (2004). The deepthought core architecture framework. In *Proceedings LREC*, pp. 1205–1208.

Chieu, H. L., H. T. Ng, and Y. K. Lee (2003). Closing the gap: Learning-based information extraction rivaling knowledge-engineering methods. In E. Hinrichs and D. Roth (Eds.), *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 216–223.

Chinchor, N. (1998). Overview of MUC-7. *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference held in Fairfax, VA, April 29-May 1, 1998*.

Ciravegna, F. (2001). Adaptive Information Extraction from Text by Rule Induction and Generalisation. *Proceedings of tht 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, 1251–1256.

Cohn, D., L. Atlas, and R. Ladner (1994). Improving generalzation with active learning. In *Machine Learning*, Volume 15(2), pp. 201–221.

Collins, M. and Y. Singer (1999). Unsupervised models for named entity classification.

Copestake, A. Implementing Typed Feature Structure Grammars. *Computational Linguistics 29*(3).

Copestake, A. (2003). Report on the Design of RMRS. Technical Report D1.1a, University of Cambridge, UK.

Copestake, A. and D. Flickinger (2000). An open-source grammar development environment and broad-coverage English grammar using HPSG. *Conference on Language Resources and Evaluation*.

Copestake, A., D. Flickinger, I. Sag, and C. Pollard (2005). Minimal Recursion Semantics. To appear.

Crouch, R. (2005). Packed rewriting for mapping semantics to KR. In *Proceedings IWCS*, Tilburg, The Netherlands.

Crysmann, B., A. Frank, B. Kiefer, H.-U. Krieger, S. Müller, G. Neumann, J. Piskorski, U. Schäfer, M. Siegel, H. Uszkoreit, and F. Xu (2002). An integrated architecture for shallow and deep processing. In *Proceedings of ACL-2002, Association for Computational Linguistics 40th Anniversary Meeting, July 7-12*, Philadelphia, USA.

Culotta, A. and J. Sorensen (2004). Dependency tree kernels for relation extraction. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*.

Cumby, C. and D. Roth (2000). Relational representations that facilitate learning. *Proc. of the International Conference on the Principles of Knowledge Representation and Reasoning*, 425–434.

Cunningham, H. (2002). GATE, a General Architecture for Text Engineering. *Computers and the Humanities 36*(2), 223–254.

Daelemans, W. and V. Hoste (2002). Evaluation of machine learning methods for natural language processing tasks. *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, 755–760.

Davidov, D., A. Rappoport, and M. Koppel (2007). Fully unsupervised discovery of concept-specific relationships by web mining. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 232–239.

Doddington, G., A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel (2004). The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. *Proceedings of LREC 2004*, 837–840.

Douthat, A. (1998). The message understanding conference scoring software users manual. *Proceedings of the 7th Message Understanding Conference (MUC-7)*.

Drożdżyński, W., H.-U. Krieger, J. Piskorski, U. Schäfer, and F. Xu (2004). Shallow processing with unification and typed feature structures — foundations and applications. *Künstliche Intelligenz 1*, 17–23.

Erk, K., A. Kowalski, S. Padó, and M. Pinkal (2003). Towards a Resource for Lexical Semantics: A Large German Corpus with Extensive Semantic Annotation. In *Proceedings of the ACL 2003*, pp. 537–544.

Etzioni, O., M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates (2005). Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence 165*(1), 91–134.

Felger, N. (2007). Portierung eines relationsextraktionssystems auf eine neue domäne. Bachelor work, the University of the Saarland, Germany.

Fillmore, C. J. (1976). Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, Volume 280, pp. 20–32.

Finkelstein-Landau, M. and E. Morin (1999). Extracting semantic relationships between terms: Supervised vs. unsupervised methods. *Workshop on Ontologial Engineering on the Global Info. Infrastructure*.

Fleischman, M., E. Hovy, and A. Echihabi (2003). Offline strategies for online question answering: Answering questions before they are asked. *Proceedings of ACL 3*, 1–7.

Frank, A., M. Becker, B. Crysmann, B. Kiefer, and U. Schäfer (2003). Integrated shallow and deep parsing: TopP meets HPSG. *Proceedings of the ACL 2003*, 104–111.

Frank, A. and K. Erk (2004). Towards an LFG syntax—semantics interface for Frame Semantics annotation. In A. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing*. LNCS, Springer.

Frank, A., H. Krieger, F. Xu, H. Uszkoreit, B. Crysmann, B. Jörg, and U. Schäfer (2005). Querying structured knowledge sources. *Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence (AAAI-05)*, 10–19.

Frank, A., H. Krieger, F. Xu, H. Uszkoreit, B. Crysmann, B. Jörg, and U. Schäfer (2006). Question answering from structured knowledge sources.

Journal of Applied Logic, Special Issue on Questions and Answers: Theoretical and Applied Perspectives 1, 29.

Freitag, D. (2000). Machine Learning for Information Extraction in Informal Domains. *Machine Learning 39*(2), 169–202.

Freitag, D. and A. K. McCallum (1999). Information extraction with hmms and shrinkage. In *Proceedings of the AAAI-99 Workshop on Machine Learning for Informatino Extraction.*

Garey, M. R. and D. S. Johnson (1979). *Computers and Intractability. A Guide to the Theory of NP-Completeness.* New York: W.H. Freeman.

Greenwood, M., M. Stevenson, Y. Guo, H. Harkema, and A. Roberts (2005). Automatically Acquiring a Linguistically Motivated Genic Interaction Extraction System. *Proceedings of the 4th Learning Language in Logic Workshop (LLL05), Bonn, Germany*.

Greenwood, M. A. and M. Stevenson (2006, July). Improving semi-supervised acquisition of relation extraction patterns. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, Sydney, Australia, pp. 29–35. Association for Computational Linguistics.

Grishman, R. (1997). Information Extraction: Techniques and Challenges. *Information Extraction (International Summer School SCIE-97)*.

Grishman, R. and B. Sundheim (1996, June). Message understanding conference - 6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen.

Guo, Y., Z. Pan, and J. Heflin (2004). An evaluation of knowledge base systems for large OWL datasets. In *Proceedings of ISWC 2003*. Springer.

Hamp, B. and H. Feldweg (1997). GermaNet-a Lexical-Semantic Net for German. *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Harabagiu, S., D. Moldovan, C. Clark, M. Bowden, J. Williams, and J. Bensley (2003). Answer Mining by Combining Extraction Techniques with Abductive Reasoning. *Proceedings of the Twelfth Text Retrieval Conference (TREC 2003)*.

Harabagiu, S., M. Pasca, and S. Maiorano (2000). Experiments with open-domain textual question answering. *Proceedings of COLING-2000*, 292–298.

Hearst, M. (1992). Automatic Acquisition of Hyponyms om Large Text Corpora. *Proceedings of the Fourteenth International Conference on Computational Linguistics*.

Hirschman, L. (1998). The Evolution of evaluation: Lessons from the Message Understanding Conferences. *Computer Speech & Language 12*(4), 281–305.

Hobbs, J., D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson (1997). FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. *Finite-State Language Processing*, 383–406.

Horridge, M. (2004). A practical guide to building OWL ontologies with the Protégé-OWL plugin. Technical report, University of Manchester.

Horrocks, I. (1998). *FaCT Reference Manual*.

Horrocks, I., U. Sattler, and S. Tobies (2000). Reasoning with individuals for the description logic SHIQ. In *Proceedings of CADE-17*. Springer.

Huffman, S. (1996). Learning to extract information from text based on user-provided examples. *Proceedings of the fifth international conference on Information and knowledge management*, 154–163.

Huttunen, S., R. Yangarber, and R. Grishman (2002a). Complexity of event structure in ie scenarios. In *Proceedings of COLING 2002: The 19th International Conference on Computational Linguistics*, Taipei, Taiwan.

Huttunen, S., R. Yangarber, and R. Grishman (2002b). Diversity of scenarios in information extraction. In *Proceedings of 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas, Canary Islands (Spain).

Inkpen, D. (2001). Graeme (2001). Building a Lexical Knowledge-Base of Near-Synonym Differences. *Proceedings of Workshop on WordNet and Other Lexical Resources (NAACL 2001), Pittsburgh*, 47–52.

Ipeirotis, P., E. Agichtein, P. Jain, and L. Gravano (2006). To search or to crawl?: towards a query optimizer for text-centric tasks. *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, 265–276.

Ireson, N., F. Ciravegna, M. E. Califf, D. Freitag, N. Kushmerick, and A. Lavelli (2005, August). Evaluating machine learning for information

extraction. *22nd International Conference on Machine Learning (ICML 2005)*.

Jijkoun, V., M. de Rijke, and J. Mur (2004). Information extraction for question answering: Improving recall through syntactic patterns. *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*.

Jones, R. (2005). *Learning to Extract Entities from Labeled and Unlabeled Text*. Ph. D. thesis, University of Utah.

Kehler, A. (1998). Learning Embedded Discourse Mechanisms for Information Extraction. *Proc. AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*.

Kim, J. and D. Moldovan (1995). Acquisition of linguistic patterns for knowledge-based information extraction. *IEEE Transactions on Knowledge and Data Engineering 7*(5), 713–724.

Klein, D. and C. Manning (2003). Accurate unlexicalized parsing. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 423–430.

Klyne, G. and J. J. Carroll (2004). Resource description framework (RDF): Concepts and abstract syntax. Technical report, W3C.

Knublauch, H., M. A. Musen, and A. L. Rector (2004). Editing description logic ontologies with the Protégé OWL plugin. In *Proc. of the International Workshop in Description Logics*.

Lavelli, A., M. Califf, F. Ciravegna, D. Freitag, C. Giuliano, N. Kushmerick, and L. Romano (2004). IE evaluation: Criticisms and recommendations. *AAAI-04 Workshop on Adaptive Text Extraction and Mining (ATEM-2004), San Jose, California*.

Li, H. (2006). Relation extraction of various complexity. Diplomarbeit, Computer Science Department, University of the Saarland, Saarbrücken, Germany.

Lin, D. (1998). Dependency-based evaluation of MINIPAR. *Workshop on the Evaluation of Parsing Systems*, 317–330.

Mann, G. and D. Yarowsky (2005, June). Multi-field information extraction and cross-document fusion. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, pp. 483–490. Association for Computational Linguistics.

McDonald, R., F. Pereira, S. Kulick, S. Winters, Y. Jin, and P. White (2005, June). Simple algorithms for complex relation extraction with applications to biomedical IE. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, pp. 491–498. Association for Computational Linguistics.

Miller, G., P. University, and C. S. Laboratory (1998). *WordNet*. MIT Press.

Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller (1993). Five papers on WordNet. Technical report, Cognitive Science Laboratory, Princeton.

Moschitti, A. and C. A. Bejan (2004, May 6 - May 7). A semantic kernel for predicate argument classification. In H. T. Ng and E. Riloff (Eds.), *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, Boston, Massachusetts, USA, pp. 17–24. Association for Computational Linguistics.

MUC-6 (1995). Proceedings of the 6th conference on message understanding.

Müller, S. and W. Kasper (2000). HPSG analysis of German. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 238–253. Berlin: Springer.

Muslea, I. (1999, July). Extraction patterns for information extraction tasks: A survey. In *AAAI Workshop on Machine Learning for Information Extraction*, Orlando, Florida.

Muslea, Ion Minton, S. and C. A. Knoblock (2002). Active + semi-supervised learning = robust multi-view learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML 2002)*, pp. 435–442.

Muslea, Ion Minton, S. and C. A. Knoblock (2003). Active learning with strong and weak views: A case study on wrapper induction. In *Proceedings of IJCAI–2003*.

Neumann, G., J. Baur, M. Becker, and C. Braun (1997). An information extraction core system for real world German text processing. *Proceedings of the fifth conference on Applied natural language processing*, 209–216.

Neumann, G. and B. Sacaleanu (2003). A Cross-language Question/Answering System for German and English. In *Proceedings of the CLEF-2003 Workshop*, Trondheim.

Neumann, G. and B. Sacaleanu (2004). Experiments on Robust NL Question Interpretation and Multi-layered Document Annotation for a Cross-

Language Question/Answering System. In *Proceedings of the Working Notes for the CLEF-2004 Workshop*, Bath, UK.

Neumann, G. and F. Xu (2003). Mining answers in German Web pages. *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, 125–131.

Newman, M., S. Strogatz, and D. Watts (2001). Random graphs with arbitrary degree distributions and their applications. *Physical Review E 64*(2), 26118.

Niles, I. and A. Pease (2001a). Origins of the Standard Upper Merged Ontology: A proposal for the IEEE standard upper ontology. In *IJCAI-2001 Workshop on the IEEE Standard Upper Ontology*.

Niles, I. and A. Pease (2001b). Towards a standard upper ontology. In C. Welty and B. Smith (Eds.), *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*.

Niles, I. and A. Pease (2003). Linking lexicons and ontologies: Mapping WordNet to the suggested upper merged ontology. In *Proceedings of the 2003 International Conference on Information and Knowledge Engineering*.

Niles, I. and A. Terry (2004). The MILO: A general-purpose, mid-level ontology. In *2004 International Conference on Information and Knowledge Engineering*, Las Vegas, NV.

Oepen, S., H. Dyvik, J. Lonning, E. Veldall, D. Beermann, J. Carroll, D. Flickinger, L. Hellan, J. Johannessen, P. Meurer, T. Nordgard, and V. Rosén (2004). Som a kapp-ete med trollet? Towards MRS-based Norwegian–English Machine Translation. In *Proceedings of the Int. Conference on Theoretical and Methodological Issues in Machine Translation*.

Pantel, P. and M. Pennacchiotti (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, 113–120.

Pease, A., I. Niles, and J. Li (2002). The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. *Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web 28*.

Pierce, D. and C. Cardie (2001). User-oriented machine learning strategies for information extraction: Putting the human back in the loop. In *IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, pp. 80–81.

Piskorski, J. and G. Neumann (2000). An intelligent text extraction and navigation system. *Proceedings of the RIAO-2000*.

Reynolds, D. (2004). *Jena 2 Inference support*.

Riezler, S., T. King, R. Kaplan, R. Crouch, J. Maxwell III, and M. Johnson (2001). Parsing the wall street journal using a Lexical-Functional Grammar and discriminative estimation techniques. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 271–278.

Riloff, E. (1993). Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of Sixteenth National Conference on Artificial Intelligence (AAAI-93)*, pp. 811–816. The AAAI Press/MIT Press.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 1044–1049. The AAAI Press/MIT Press.

Romano, L., M. Kouylekov, I. Szpektor, I. Dagan, and A. Lavelli (2006). Investigating a Generic Paraphrase-based Approach for Relation Extraction. *Proceedings of EACL*.

Sager, N., C. Friedman, and S. Margaret (1987). *Medical Language Processing: Computer Management of Narrative Data*. Addison-Wesley.

SAIC.

Salton, G. (1991). Developments in Automatic Text Retrieval. *Science 253*(5023), 974.

Salton, G. and M. McGill (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc. New York, NY, USA.

Schäfer, U. (2004). Using XSLT for the integration of deep and shallow natural language processing components. In *Proceedings of the ESSLLI 2004 workshop on Combining Shallow and Deep Processing for NLP*, pp. 31–40.

Schäfer, U. (2007, 6). *Integrating Deep and Shallow Natural Language Processing Components - Representations and Hybrid Architectures*, Volume 22 of *Saarbrücken Dissertations in Computational Linguistics and Language Technology*. Saarbrücken, Germany: DFKI GmbH and Computational Linguistics Department, Saarland University.

Soderland, S. (1999). Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning 34*(1), 233–272.

Soderland, S., D. Fisher, J. Aseltine, and W. Lehnert (1995). CRYSTAL: Inducing a conceptual dictionary. In C. Mellish (Ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, San Francisco, pp. 1314–1319. Morgan Kaufmann.

Soderland, S. and W. Lehnert (1995). Learning domain-specific discourse rules for information extraction. In *Proceedings of the AAAI 1995 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, Stanford, CA, pp. 143–148.

Spreyer, K. and A. Frank (2005). Projecting RMRS from TIGER dependencies. University of the Saarland and DFKI, submitted.

Stevenson, M. and M. Greenwood (2005). A Semantic Approach to IE Pattern Induction. *Ann Arbor 100*.

Stevenson, M. and M. A. Greenwood (2006, July). Comparing information extraction pattern models. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, Sydney, Australia, pp. 12–19. Association for Computational Linguistics.

Subirats, C. and H. Sato (2004). Spanish FrameNet and FrameSQL. In *Proceedings of the LREC Workshop on Building Lexical Resources from Semantically Annotated Corpora*.

Suchanek, F. M., G. Ifrim, and G. Weikum (2006, July). Leila: Learning to extract information by linguistic analysis. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, Sydney, Australia, pp. 18–25. Association for Computational Linguistics.

Sudo, K., S. Sekine, and R. Grishman (2001). Automatic pattern acquisition for japanese information extraction. In *HLT '01: Proceedings of the first international conference on Human language technology research*, Morristown, NJ, USA, pp. 1–7. Association for Computational Linguistics.

Sudo, K., S. Sekine, and R. Grishman (2003). An improved extraction pattern representation model for automatic IE pattern acquisition. *Proceedings of ACL 2003*, 224–231.

Surdeanu, M., S. Harabagiu, J. Williams, and P. Aarseth (2003). Using predicate-argument structures for information extraction. *Proceedings of*

ACL 2003.

Swier, R. and S. Stevenson (2004). Unsupervised semantic role labelling. *Proc. of the 2004 Conf. on EMNLP*, 95–102.

Tapanainen, P. and T. Jarvinen (1997). A non-projective dependency parser. *Proceedings of the 5th Conference on Applied Natural Language Processing*, 64–71.

Thompson, C. A., M. E. Califf, and R. J. Mooney (1999, June). Active learning for natural language parsing and information extraction. In *Proceedings of the Sixteenth International Machine Learning Conference (ICML-99)*, Bled, Slovenia, pp. 406–414.

Tsujii, J. (2000). Generic NLP technologies: language, knowledge and information extraction. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 12–22.

Turney, P. (2006). Expressing Implicit Semantic Relations without Supervision. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, 313–320.

Uszkoreit, H. (2002). New chances for deep linguistic processing. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02), August 24 - September 1*. Morgan Kauffmann Press.

Uszkoreit, H. (2007). Invited talk: Methods and applications for relation detection – potential and limitations of automatic learning in ie. *2007 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE 2007)*.

Uszkoreit, H., B. Jörg, and G. Erbach (2003). An ontology-based knowledge portal for language technology. In *Proc. of ENABLER/ELSNET WS* Intern. Roadmap for Language Resources.

Uszkoreit, H., G. Neumann, S. Oepen, S. Spackman, R. Backofen, S. Busemann, A. Diagne, E. Hinkleman, W. Kasper, B. Kiefer, et al. (1994). DISCO: an HPSG-based NLP system and its application for appointment scheduling. *Proceedings of the 15th conference on Computational linguistics-Volume 1*, 436–440.

Uszkoreit, H. and F. Xu (2007). Semantic model for information extraction. DFKI, forthcoming.

Voorhees, E. (2003). Overview of the TREC 2003 Question Answering Track. *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*.

Wahlster, W. (2000). *Verbmobil:: Foundations of Speech-to-speech Translation*. Springer.

Xu, F. (2003). *Multilingual WWW — Modern Multilingual and Cross-lingual Information Access Technologies*, Chapter 9, pp. 165–184. Kluwer Academic Publishers.

Xu, F. (2004). Linking flat predicate argument structure. Research Report RR-04-04, DFKI, German Research Center for Artificial Intelligence, Germany.

Xu, F. and H.-U. Krieger (2003, 9). Integrating shallow and deep nlp for information extraction. In *Proceedings of RANLP 2003*, Borovets, Bulgaria.

Xu, F., D. Kurz, J. Piskorski, and S. Schmeier (2002). A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping. In *the third international conference on language resources and evaluation (LREC)*, Spain, pp. 224–230.

Xu, F. and H. Uszkoreit (2007, May). Minimally supervised learning of relation extraction rules using semantic seeds. A seminar talk at the National Center for Text Mining (NaCTeM).

Xu, F., H. Uszkoreit, and H. Li (2006, July). Automatic event and relation detection with seeds of varying complexity. In *Proceedings of AAAI 2006 Workshop Event Extraction and Synthesis*, Boston.

Xu, F., H. Uszkoreit, and H. Li (2007). A seed-driven bottom-up machine learning framework for extracting relations of various complexity. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL07)*, 584–591.

Yangarber, R. (2001). *Scenarion Customization for Information Extraction*. Dissertation, Department of Computer Science, Graduate School of Arts and Science, New York University, New York, USA.

Yangarber, R. and R. Grishman (1997). Customization of Information Extraction Systems. In P. Velardi (Ed.), *International Workshop on Lexically Driven Information Extraction*, Frascati, Italy, pp. 1–11. Università di Roma.

Zelenko, D. and C. Richardella (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research 3*(6), 1083–1106.

Zelle, J. and R. Mooney (1994). Combining top-down and bottom-up methods in inductive logic programming. *Proceedings of the Eleventh Interna-*

tional Conference on Machine Learning 343351.

Zhang, Y. and V. Kordoni (2006). Automated deep lexical acquisition for robust open texts processing. *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*.

Zhang, Y., S. Oepen, and J. Carroll (2007). Efficiency in unification-based n-best parsing. *Proceedings of the Tenth International Conference on Parsing Technologies*, 48–59.

Zhao, S. and R. Grishman (2005). Extracting Relations with Integrated Information Using Kernel Methods. *Ann Arbor 100*.

Zheng, Z. (2002). AnswerBus Question Answering System. *Ann Arbor 1001*, 48109.