Effective Distant Supervision for End-To-End Knowledge Base Population Systems

Benjamin Roth

Dissertation zur Erlangung des Grades des Doktors der Ingenieurwissenschaften der Naturwissenschaftlich-Technischen Fakultäten der Universität des Saarlandes

Thesis for obtaining the title of Doctor of Engineering of the Faculties of Natural Sciences and Technology of Saarland University

Saarbrücken, Germany, 2014

Dean:	Prof. Dr. Markus Bläser
	Naturwissenschaftlich-Technische Fakultät I - Mathematik und Informatik - (Faculty of Mathematics and Computer Science) Universität des Saarlandes
Colloquium:	December 22, 2014 Universität des Saarlandes Saarbrücken

Examination Board:

Chairman:	Prof. Dr. Raimund Seidel Universität des Saarlandes
First Reviewer:	Prof. Dr. Dietrich Klakow Universität des Saarlandes
Second Reviewer:	PD. Dr. Fabian Suchanek Télécom ParisTech
Committee Member: (Akademischer Mitarbeiter)	Dr. Michael Wiegand Universität des Saarlandes

Abstract

The growing amounts of textual data require automatic methods for structuring relevant information so that it can be further processed by computers and systematically accessed by humans. The scenario dealt with in this dissertation is known as *Knowledge Base Population (KBP)*, where relational information about entities is retrieved from a large text collection and stored in a database, structured according to a prespecified schema. Most of the research in this dissertation is placed in the context of the KBP benchmark of the Text Analysis Conference (TAC KBP), which provides a test-bed to examine all steps in a complex end-to-end relation extraction setting.

In this dissertation a new state of the art for the TAC KBP benchmark was achieved by focussing on the following research problems: (1) The KBP task was broken down into a modular pipeline of sub-problems, and the most pressing issues were identified and quantified at all steps. (2) The quality of semi-automatically generated training data was increased by developing noise-reduction methods, decreasing the influence of false-positive training examples. (3) A focus was laid on fine-grained entity type modelling, entity expansion, entity matching and tagging, to maintain as much recall as possible on the relational argument level. (4) A new set of effective methods for generating training data, encoding features and training relational classifiers was developed and compared with previous state-of-the-art methods.

Kurzzusammenfassung

Die wachsende Menge an Textdaten erfordert Methoden, relevante Informationen so zu strukturieren, dass sie von Computern weiterverarbeitet werden können, und dass Menschen systematisch auf sie zugreifen können. Das in dieser Dissertation behandelte Szenario ist unter dem Begriff *Knowledge Base Population (KBP)* bekannt. Hier werden relationale Informationen über Entitäten aus großen Textbeständen automatisch zusammengetragen und gemäß einem vorgegebenen Schema strukturiert. Ein Großteil der Forschung der vorliegenden Dissertation ist im Kontext des *TAC KBP* Vergleichstests angesiedelt. Dieser stellt ein Testumfeld dar, um alle Schritte eines anfragebasierten Relationsextraktions-Systems zu untersuchen.

Die in der vorliegenden Dissertation entwickelten Verfahren setzen einen neuen Standard für *TAC KBP*. Dies wurde durch eine Schwerpunktsetzung auf die folgenden Forschungsfragen erreicht: Erstens wurden die wichtigsten Unterprobleme von KBP identifiziert und die jeweiligen Effekte genau quantifiziert. Zweitens wurde die Qualität von halbautomatischen Trainingsdaten durch Methoden erhöht, die den Einfluss von falsch positiven Trainingsbeispielen verringern. Drittens wurde ein Schwerpunkt auf feingliedrige Typmodellierung, die Expansion von Entitätennamen und das Auffinden von Entitäten gelegt, um eine größtmögliche Abdeckung von relationalen Argumenten zu erreichen. Viertens wurde eine Reihe von neuen leistungsstarken Methoden entwickelt und untersucht, um Trainingsdaten zu erzeugen, Klassifizierungsmerkmale zu kodieren und relationale Klassifikatoren zu trainieren.

Acknowledgements

During the years of my PhD I was supported by several people, who not only made this work an invaluable learning experience, but also an enjoyable journey and an opportunity of personal growth. I am very thankful for their company in the successes and difficult challenges of this journey, and for making this ambitious undertaking an experience that I would not want to miss.

First and foremost, my thanks go to my advisor Dietrich Klakow who supported me constantly, created an atmosphere of open discussion, and who knew how to ask precise and constructive questions that were necessary so that initial ideas could take on a concrete shape. He also took a lot of time to read earlier versions of this thesis and I am thankful for the constructive criticism that helped to improve the final version. Also, during the time of my PhD, teaching the seminars on statistical methods in NLP together with him was one of the most enjoyable duties that I had.

I am very grateful for the generous support by a Google Europe Fellowship in Natural Language Processing that made my PhD possible. Additionally, I greatly profited from the inspiring time that I spent during an internship at Google Research Zurich, and I am especially indebted to Keith Hall and Enrique Alfonseca, with whom I always had a fruitful exchange of ideas and discussions of my research, even after the internship at Google was long over.

I received very important support from my colleagues in the Spoken Language System lab at Saarland University. The constant exchange of ideas with them was inspiring and kept me motivated to pursue my research goals. I greatly profited both from their feedback as well as from their help with many practical issues. I am especially indebted to Michael Wiegand for the many times he helped me with developing ideas, proof-reading papers and parts of this thesis, and for the constructive collaboration on relation extraction in the food domain. A special thanks goes to Tassilo Barth for the great teamwork in turning dozens of scripts into a solid software system for relation extraction. A big thanks goes to Martin Gropp, for providing the great demo interface for the relation extraction system, and for being an invaluable help when I was struggling with the dark corners of shell scripting, linux networks and servers of all kinds. I'm very thankful to Grzegorz Chrupala and Mittul Singh for working together in getting the relation extraction system ready for the submissions to the TAC

competition.

The years during my PhD would not have been what they were without Joanna, who shared with me her creativity and with whom I could enjoy culture and life. I am truly thankful to her for being so reliable and supportive. Heartfelt thanks also go to my family, and especially to my father for proofreading the thesis for stylistic improvement; to my mother, my sister and my brother, for being there for me and for supporting me in everything that I do.

Contents

List of Figures

List of Tables

1. Introduction

The immense growth of available data on the web has long been recognized as a main motivation for better data analysis and searching techniques. Yet, not only has the sheer amount of data increased, but also the number of use-cases and users that would profit from better access to the information contained in the vast amount of textual data available. The main difficulty in searching large amounts of textual data lies in the fact that it is *unstructured*: information can be expressed in free form with no restrictions on complexity or scope. The opposite would be a *structured* data repository, such as a data base with a number of tables listing information according to a specified schema.

There are several ways to access unstructured data: one form that is immediately accessible to humans is to write queries freely specifying their information needs and to display the relevant information in the original unstructured form. This is the strategy usually employed by web search engines such as Google. Another strategy that both makes information more accessible to humans and also allows computers to process the data further, e.g. as part as of a larger information system, is to bring the relevant parts of the unstructured information into a structured form of data tables with pre-specified relationships.

More and more scenarios arise which demand the tabularization of unstructured information from text. They range from extending encyclopedic knowledge with structured data, e.g. extending Wikipedia infoboxes (Hoffmann et al., 2010), over automatic analysis and indexing of research papers, such as in the bio-medical domain (Segura-Bedmar et al., 2013), to an increasing usage of automated and quantitative methods in the humanities, such as the extraction of relationships between fictional characters in literary works (Elson et al., 2010). This wide range of novel use-cases requires reliable, robust and well understood methods for relation extraction.

Most of the research in this dissertation is placed in the context of the Knowledge Base Population benchmark of the Text Analysis Conference (TAC KBP) organized by the National Institute for Standards and Technology (NIST).[1] The problem setting for Knowledge Base Population is to search a large text collection for relevant information

[1] `http://www.nist.gov/tac/about/index.html`

given a set of queries, and to structure the retrieved information according to a pre-specified relational schema. This setup together with its extensive manually created evaluation data provides an ideal setup to examine all steps in a complex end-to-end relation extraction setting.

While much of the existing research focuses on single steps and methods in a relation extraction scenario, we aim at studying the end-to-end setting with all the aspects that are involved. The goal of this dissertation is to investigate *what really matters* for performing automatic relation extraction on a level that advances the state of the art. Our approach is therefore to commit to an evaluation setting that is well-justified and well-defined to measure progress in Knowledge Base Population and to seek to devise algorithms approaching that goal.

This dissertation is structured as follows: Chapter 2 gives an overview of previous research in relation extraction and shows the motivation for different relation extraction tasks as well as underlying similarities and distinguishing features. Chapter 2 also defines the problem dealt with in this dissertation, Knowledge Base Population, as *fact*-centered relation extraction with a defined schema. Chapter 3 turns to the problem that for relation extraction tasks often only very little or no training data is available. As this renders supervised training unfeasible, *distant supervision* has been proposed, a semi-supervised scheme that uses an initial knowledge base to generate (imperfect) training examples. In Chapter 3, we describe and quantify the types of errors introduced by this training scheme and discuss existing techniques of mitigating some of the inherent problems. In Chapter 4, we propose two models for improved prediction with noisy data: a feature-based generative model and a discriminative model based on ranking constraints. We extensively evaluate those models (and their combination) and compare them with several state-of-the-art baselines.

In Chapter 5, we widen the focus and lay out the architecture of the end-to-end *RelationFactory* relation extraction system, which was developed in the context of this dissertation and released as open source. We evaluate, for each module, the specific impact on overall performance and motivate the resulting design choices. The pipeline is divided into two stages: (1) a recall oriented *candidate generation* stage, for retrieving and detecting all contexts that potentially contain information related to a query entity and (2) a precision-oriented *candidate validation* stage that functions as a filter

and checks whether the query-related information is relevant according to the specified information need. Query expansion, retrieval and named entity tagging are central to the candidate generation stage are. We compare several schemes developed for those tasks and give a thorough analysis of the amount of recall missed during the candidate generation stage. In the second half of Chapter 5 we discuss the relational predictors developed for the candidate validation stage. Apart from the distant supervision predictors already discussed in Chapter 4, the most notable component in this stage is a set of support vector machine classifiers. We extensively experimented with different settings regarding training data, parameter tuning and feature sets. The classification experiments show the advantage of shallow skip-n-gram features when compared with a more brittle state-of-the-art feature set based on dependency parsing. In Chapter 6 we discuss the possibility to extend pipelined approaches by a sequence-labeler, that can undo decisions made by earlier models. We report initial experiments with promising results.

Chapter 7 gives an overview of the 18 systems that participated in the TAC KBP 2013 evaluation campaign and discusses the different research approaches taken. *RelationFactory*, which was top-ranked in this benchmark, is compared with other systems some of which are similar in certain design aspects (e.g. the Stanford and NYU systems (Angeli et al., 2013; Grishman, 2013)) and others which are dissimilar in the overall approach (e.g. the UMass system (Singh et al., 2013)). Chapter 8 takes a closer look at the problem of detecting relational arguments when they are not of standard entity types. For this purpose, we turn to the setting of relation prediction in the food domain, where no training data for learning a sequence label tagger exists. We introduce semi-supervised graph-based methods for propagating labeling information with as little as 10 manual seeds per category. We show that this minimally supervised categorization is beneficial to relation prediction and is competitive with another more resource-intensive method using GermaNet (Hamp and Feldweg, 1997), the German WordNet (Miller, 1995). In Chapter 9 we discuss issues pertaining to the re-usability of current knowledge base evaluation resources, and propose a setting which would evaluate the same system aspects while allowing for better re-usability.

The main contributions of this dissertation are:

- A distant supervision noise reduction model based on ranking constraints.

- A feature-based extension to a generative noise reduction model.
- A new state-of-the-art in performance for relation extraction and knowledge-base population.
- A detailed breakdown of the impact of different modules and design choices on end-to-end performance.
- A shallow feature set that outperforms the previous state-of-the-art based on dependency parses.
- A minimally supervised graph-based method for type-clustering of non-standard named entities.

Parts of this dissertation have been published in the following research papers:

- Benjamin Roth, Tassilo Barth, Grzegorz Chrupała, Martin Gropp, Dietrich Klakow. RelationFactory: A Fast, Modular and Effective System for Knowledge Base Population. EACL 2014.
- Michael Wiegand, Benjamin Roth and Dietrich Klakow. Automatic Food Categorization from Large Unlabeled Corpora and Its Impact on Relation Extraction. EACL 2014.
- Benjamin Roth, Tassilo Barth, Michael Wiegand, Mittul Singh, Dietrich Klakow. Effective Slot Filling Based on Shallow Distant Supervision Methods. NIST Text Analysis Conference 2013.
- Benjamin Roth, Dietrich Klakow. Combining Generative and Discriminative Model Scores for Distant Supervision. EMNLP 2013.
- Benjamin Roth, Tassilo Barth, Michael Wiegand, Dietrich Klakow. A Survey of Noise Reduction Methods for Distant Supervision. CIKM 2013 Workshop on Knowledge Extraction (AKBC).

2. Overview of Relation Extraction and Knowledge Base Population

Relation extraction is a wide topic, and depending on the goal or focus of research, widely varying approaches are taken. In this section, we give an overview of several aspects that determine the objects of study for a particular method. We will also define what we understand by *Knowledge Base Population* in the context of this dissertation and introduce two dimensions that are relevant in the context of relational analysis of text: First, whether the (relation analysis) task is defined in *linguistic* terms, or whether the (relation extraction) task is defined in terms of *facts*, irrespective of what linguistic form was used to express those facts. Second, whether the relations considered fall into a pre-specified schema, or whether constructing such a schema is itself part of the task. Furthermore, in addition to the previous distinction, we use the term *ontology* to describe all knowledge repositories that contain *meta*-knowledge, i.e. information about rules on how to combine other knowledge.

In the following chapter, we will position several research tasks along those lines. For example, *automatic paraphrase clustering* (Lin and Pantel, 2001a) would be a linguistically motivated task with an open schema, since it is defined on linguistic building blocks (short phrases) irrespective of particular instantiations and has no underlying specified schema. *Knowledge Base Population* on the opposite end aims at extracting *content* from a text describing real world events, according to a predefined schema. The *Knowledge Base Population* approach, falling into the category of extracting real world facts given a defined schema, can be formulated in the following way: *Given a specific information need, expressed as a set of relations of interest, find the relevant information in a large amount of unstructured text, and return it in a structured form.*

The rest of this chapter is organized as follows: In Section 2.1, we discuss the difference between relation analysis tasks that are defined in linguistic terms and those that are motivated by real world information needs. In Section 2.2, we illustrate this distinction by contrasting two evaluation benchmarks: the earlier ACE benchmark, and the recent KBP slotfilling. The difference between open schema and closed schema relation extraction is discussed in Section 2.3. Other aspects of relation extraction are

covered in a more comprehensive literature overview in Section 2.4.

2.1. Linguistic Relations, Facts and Ontologies

To clarify the scope of work in this dissertation, we first make a distinction between *linguistic relations*, *facts* and *ontologies*. We denote as *linguistic relations* those relations that hold between referents in a text or discourse. Linguistic relations do not necessarily have a grounding or references outside the discourse (text) in which they are expressed (e.g. in the real world). Linguistic relations allow for abstractions suitable for text-related tasks like textual inference (Tatu and Moldovan, 2005; Burchardt et al., 2007, 2009) or discourse analysis (Ruppenhofer et al., 2010; Louis and Nenkova, 2012; Wang et al., 2010). Semantic role labels, such as PropBank (Kingsbury and Palmer, 2002) or FrameNet annotations[2] (Baker et al., 1998; Burchardt et al., 2006), as well as discourse annotations (Mann and Thompson, 1988; Hovy and Maier, 1995; Prasad et al., 2008; Petukhova et al., 2011) also denote linguistic relations. For example, in the sentence *"I like to have plants on my desk"* there holds a linguistic relation `on-top-of` between the discourse referents `plants` and `my desk`. This type of relation is distinct from both factual or ontological relations as defined below.

Factual relations or *facts* hold between entities with a grounding outside of a discourse expressing them (they can even exist independently of any text). They hold permanently or can be linked to a specific point in time or period of time. Factual relations are typically stored in knowledge bases, such as data repositories that contain information about persons, products, organizations and other domains of interest. Factual relations are the object of relation extraction as understood in our work. For example, from the text *"Glasgow is west of Edinburgh"* a fact `west-of(Glasgow, Edinburgh)` between the two non-linguistic geographical entities Glasgow and Edinburgh can be inferred. While in the above example, the connection between linguistic form and factual content is relatively direct, often many facts can be inferred from the same text: *"Edinburgh is the capital of Scotland"* supports the facts `capital-of(Edinburgh, Scotland)` and `located-in(Edinburgh, Scotland)`. Obviously, a connection can be established between the linguistic form, its respective linguistic relations and expressed facts. However, this connection is often based on inferences. Also, while there might be

[2] `https://framenet.icsi.berkeley.edu`

a correlation between the linguistic form and a factual content, rarely is there a strict one-to-one correspondence. The mapping between linguistic or semantic relations and facts can also be challenging when linguistic relations are already abstractions (e.g. subject-verb-object tuples) that are devoid of additional contextual signals (such as words that add a certain aspect to a linguistic construction) necessary for recognizing facts. Semantic representations such as FrameNet are often too coarse generalizations.

Ontologies (Miller, 1995; Niles and Pease, 2001) capture generalities that hold within classes of entities, relations and their typical arguments. Ontologies contain generalizations and rules that allow for combining knowledge. They may also contain background knowledge, i.e. facts that are assumed to be generally known and are therefore not linguistically expressed. Therefore, ontologies target at covering the rules of inference, i.e. the principles of obtaining new facts from a given set of facts by combination, filtering and simplification. Automatic ontology learning approaches (Buitelaar et al., 2005; Poon and Domingos, 2010), most notably the efforts to extend WordNet (Snow et al., 2006), typically aggregate over occurrences of recurring linguistic relational patterns, rather than extract single relational instances or novel facts.

2.2. Automatic Content Extraction (ACE) and Knowledge Base Population (KBP)

Text	**Annotation**
your priest	`Per-Social.Lasting(‘‘your’’, ‘‘your priest’’)`
a guy I knew	`Per-Social.Lasting(‘‘a guy I knew’’, ‘‘I’’)`
He and a hunting partner	`Per-Social.Lasting(‘‘He’’, ‘‘a hunting partner’’)`
state-controlled banks	`Part-Whole.Subsidiary(‘‘banks’’, ‘‘state’’)`
the top of the mountain	`Part-Whole.Geo(‘‘the top of the mountain’’, ‘‘the mountain’’)`
the lobby of the hotel	`Part-Whole.Geo(‘‘the lobby of the hotel’’, ‘‘the hotel’’)`

Table 1: Text snippets and gold annotations (ground truth) according to the ACE 2008 guidelines.

The difference between linguistic relations and facts is mirrored in the two series of benchmark tasks organized by NIST: First, the **Automatic Content Extraction**

(ACE) [3] task was yearly organized from 1999 to 2008 (with the exception of 2006). ACE focused on recognizing all instances of a number of linguistic relations, i.e. in a given set of texts, relationships had to be recognized between all participants in the discourse. Relation detection in ACE was only concerned with the textual level, it was not necessary that relational arguments be grounded outside a discourse, they were not connected to any knowledge base. Table 1 gives example snippets of sentences together with their relational annotations from the ACE 2008 task specifications and annotation guidelines[4].

Even without knowing the exact meaning of the recognized relation, it is evident that the immediate result of such an annotation is not usable as a knowledge base independently of the annotated text. Moreover, the relations are abstractions from linguistic forms and often too general to be of interest for a knowledge base with a specific use case. For example, according to the guidelines:
"[Per-Social.Lasting] captures relationships that meet the following conditions:

1. *The relationship must involve personal contact (or a reasonable assumption thereof).*

2. *There must be some indication or expectation that the relationship exists outside of a particular cited interaction."*

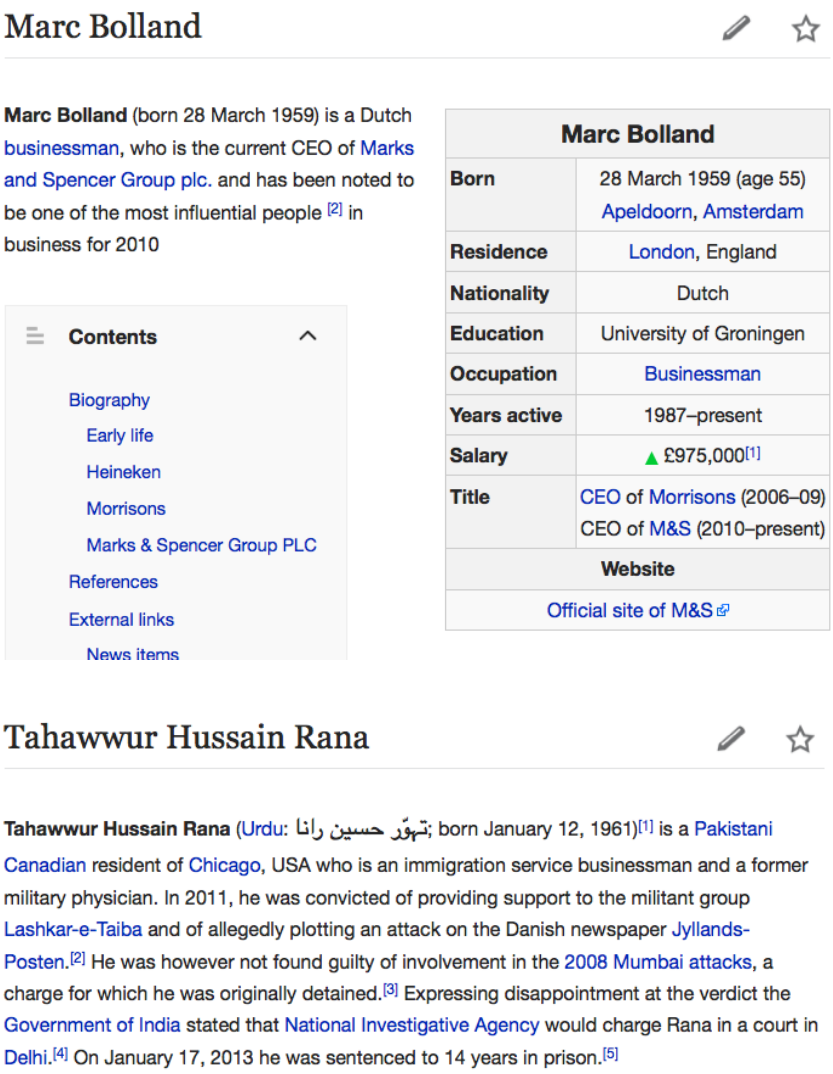
However, in most use cases one is probably interested in more specific relationships. Even under the assumption that one is interested in a knowledge base covering exactly the same set of relations, a meaningful linking to entities grounded outside of the discourse does not seem possible for most of the cases. These problems are inherent in linguistic relational annotations.

Second, the follow-up task to ACE has been the **Knowledge Base Population (KBP)** track[5] annually organized in association with the Text Analysis Conference (TAC), starting in 2009. TAC KBP starts with a *knowledge base* in mind that contains relations between entities. The 42 relations considered are derived from popular infobox schemata (tables of typical information for certain entity types) for Wikipedia

[3] `http://www.itl.nist.gov/iad/mig/tests/ace/`

[4] `http://projects.ldc.upenn.edu/ace/docs/English-Relations-Guidelines_v6.2.pdf`

[5] http://www.nist.gov/tac/tracks/index.html



Marc Bolland

Marc Bolland (born 28 March 1959) is a Dutch businessman, who is the current CEO of Marks and Spencer Group plc. and has been noted to be one of the most influential people [2] in business for 2010

Contents

- Biography
  - Early life
  - Heineken
  - Morrisons
  - Marks & Spencer Group PLC
- References
- External links
  - News items

Marc Bolland	
Born	28 March 1959 (age 55) Apeldoorn, Amsterdam
Residence	London, England
Nationality	Dutch
Education	University of Groningen
Occupation	Businessman
Years active	1987–present
Salary	▲ £975,000[1]
Title	CEO of Morrisons (2006–09) CEO of M&S (2010–present)
Website	
Official site of M&S	

Tahawwur Hussain Rana

Tahawwur Hussain Rana (Urdu: تہوّر حسین رانا; born January 12, 1961)[1] is a Pakistani Canadian resident of Chicago, USA who is an immigration service businessman and a former military physician. In 2011, he was convicted of providing support to the militant group Lashkar-e-Taiba and of allegedly plotting an attack on the Danish newspaper Jyllands-Posten.[2] He was however not found guilty of involvement in the 2008 Mumbai attacks, a charge for which he was originally detained.[3] Expressing disappointment at the verdict the Government of India stated that National Investigative Agency would charge Rana in a court in Delhi.[4] On January 17, 2013 he was sentenced to 14 years in prison.[5]

Figure 1: Wikipedia articles (snippets) with infobox (top) and without infobox (bottom).

entries about persons or organizations. This way, the relations express a real information need rather than being linguistic abstractions. Figure 1 shows Wikipedia articles with and without infoboxes. Note that even if infoboxes are present, they can be incomplete: In the given example, the fact that Bolland was chief operating officer for Heineken until 2005 is missing from the infobox, but is mentioned later in the Wikipedia article.

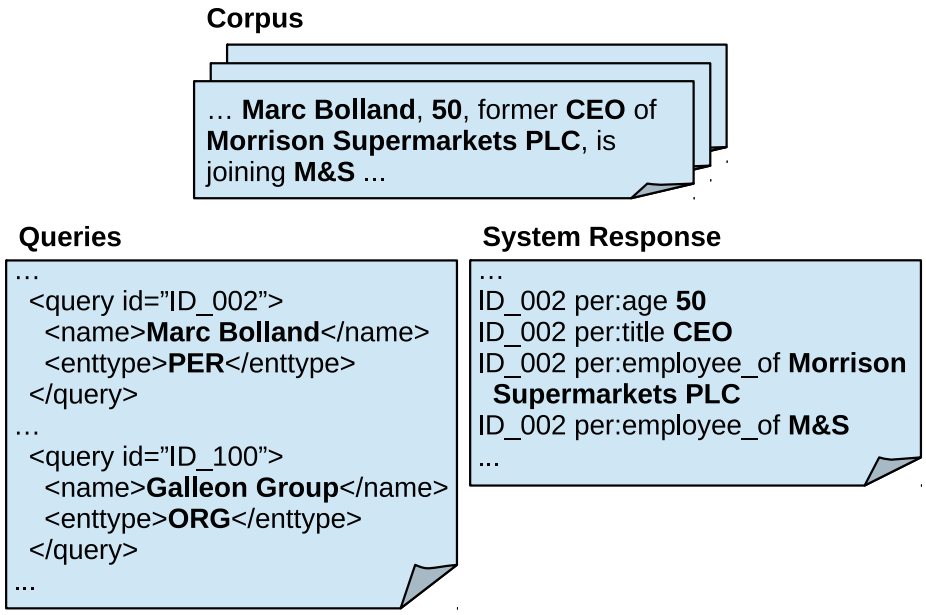


Figure 2: TAC KBP: Given a set of queries, return a correct, complete and non-redundant response with relevant information extracted from the text corpus.

To give an example of how the relations in TAC are different in their entity grounding and specificity from those in ACE, consider the social relations (relations between persons) in TAC (in contrast to the [Per-Social.Lasting] relation mentioned above for ACE). There are 5 social relations in TAC: *parent, child, sibling, spouse, other family.* These relations are defined in terms of states of the world, not linguistic forms, e.g. for the *other family* relation:
"Family other than siblings, parents, children, and spouse (or former spouse). Correct fillers for this slot include brothers-in-law, sisters-in-law, grandparents, grandchildren, cousins, aunts, uncles, etc."[6] A summary of the TAC KBP relation definitions can be found in Appendix A.1 on page 130.

Figure 2 shows a simplified example of a TAC KBP slot filling query, corpus and expected result. The task is set up as a retrieval task, specifying a query entity of given type (person or organization) which is grounded by a specified disambiguation document (the pointers for the disambiguation information have been left out in Figure

[6] `http://surdeanu.info/kbp2013/TAC_2013_KBP_Slot_Descriptions_1.0.pdf`

2 for simplicity). For such a query, all facts (from the set of the specified 42 relations) have to be retrieved if they are expressed in the so-called *source corpus* [7]. Since the relations are *factual relations*, grounded in entities outside of specific discourses, they could in theory be evaluated for truth independently of their textual occurrences – however, for several practical reasons it is assumed that a fact is true if and only if any of the documents in the source corpus states it so. Evaluation is less bound to linguistic form than it has been the case for ACE: any inference (including the use of world-knowledge) or signals that unambiguously pose the truth of a certain fact are valid support for the existence of that fact.[8]

2.3. Open vs. Closed Schema

Open schema relation extraction refers to approaches where fact tuples are the result of clustering surface forms, i.e. sequences of words or parse-spans between relational arguments. Often, surface forms are clustered simultaneously with the entity types of the arguments (Banko et al., 2007; Bollegala et al., 2010). The aim is to automatically find a generally useful partitioning and granularity. Paraphrase finding and clustering, as done in the DIRT system (Lin and Pantel, 2001a), is a linguistically motivated open schema relational analysis task. A range of open information extraction systems (OpenIE) aim at extracting facts from such clusters of surface forms. Universal Schema (Riedel et al., 2013) aims at finding an lower-dimensional vector representation (*embedding*) from un-annotated data. Instead of providing a fixed clustering, some additional flexibility is retained since similarity between all vectors can be compared with such a representation in a smooth manner. Similar embeddings are learned by the model of Weston et al. (2013). Here arguments are vectors and relations are matrices that translate one argument into another.

To illustrate the output of a typical OpenIE system, we include below an example given by Banko et al. (2007). Here, the system outputs a set consisting of 4 argument-surface tuples which were clustered together as being synonymous. A human annotator

[7] The source corpus contains more than 2 million documents, including 1,000,257 newswire documents from the Gigaword corpus and 1,099,062 documents from web text and discussion fora

[8] This does not mean that systems need to attempt *logical* inference – rather, it gives them the freedom to use any signals present in the text, whether they support a fact directly or indirectly.

may conclude that the first tuple in fact is not synonymous to the others, and mark it accordingly.

*(Bletchley Park, *was location of*, Station X)
(Bletchley Park, *being called*, Station X)
(Bletchley Park, *known as*, Station X)
(Bletchley Park, *codenamed*, Station X)

Evaluation is a particular challenge for open information extraction: Usually sets or pairs of paraphrases are presented to human raters, who should judge whether the paraphrases are reasonable and synonymous (Banko et al., 2007); or, human annotators are asked to find paraphrases to a given surface form (the *seed*) and those paraphrases are compared to the other surface forms in the automatically found cluster into which the seed falls (Lin and Pantel, 2001a). These evaluation scenarios are often inherently vague, and there is usually no evaluation of the usefulness regarding relation extraction tasks where *specific information needs* would be clearly specified in advance. A probable use case of open schema extractions may therefore lie rather in exploratory scenarios than in applications targeting a pre-specified goal.

As the key characteristic of open relation extraction is that it is very general, the hope is that it may be an aid in a range of different settings. A further advantage is that no annotated training data is needed, the final clustering is only dependent on the input data and clustering scheme, and not on requirements of a specific task. It can therefore be said that by its generality and lack of task-specificity open schema relation extraction has similar strengths and weaknesses as linguistic relation annotation does.

Closed schema approaches start with identifying the relations of interest, see Figure 3 for the schema used in TAC KBP Slot Filling. This can be done by describing the semantics/meaning of the relations (this corresponds to what is called *intension of a concept* in logic or semiotics), or by extrapolating from a sample of seed instances/facts of these relations (what might be called a partial *extension of a concept*). From the intensional descriptions one can engineer tailored extractors, e.g. compose surface patterns manually. The extensional seed facts can be the input to automatic training or generalization.

Using seed facts together with a text corpus to learn textual extractor is known as *distant supervision* and will be discussed in length in the subsequent chapters. Note

per:age	org:alternate_names
per:alternate_names	org:city_of_headquarters
per:cause_of_death	org:country_of_headquarters
per:charges	org:date_dissolved
per:children	org:date_founded
per:cities_of_residence	org:founded_by
per:city_of_birth	org:member_of
per:city_of_death	org:members
per:countries_of_residence	org:number_of_employees_members
per:country_of_birth	org:parents
per:country_of_death	org:political_religious_affiliation
per:date_of_birth	org:shareholders
per:date_of_death	org:stateorprovince_of_headquarters
per:employee_or_member_of	org:subsidiaries
per:origin	org:top_members_employees
per:other_family	org:website
per:parents	
per:religion	
per:schools_attended	
per:siblings	
per:spouse	
per:stateorprovince_of_birth	
per:stateorprovince_of_death	
per:statesorprovinces_of_residence	
per:title	

Figure 3: TAC KBP relational schemata for the entity types *person* and *organization*.

that in TAC KBP, both intensional definitions (by the task guidelines) as well as extensions in the form of Wikipedia infoboxes (from which the definitions are derived) are provided.[9] While closed schema extractors aim exactly at the desired questions, an obvious disadvantage is the need for training data (or the tailoring of the system towards those relations).

[9]It is interesting, however, that for the Wikipedia infoboxes, no definitions exist (apart from their relation names themselves), as it is the case with most relations of the publicly available Freebase knowledge base (`http://www.freebase.com/`). In other words, the intensional definitions for TAC KBP are the result of a human cognitive effort of generalization, i.e. *induction*, from the extension. Creating a system for relational prediction would correspondingly be a *deductive* effort.

2.4. Literature Overview

Chronologically, work on extracting structured information from text started with the (semi-)automatic construction of ontologies (Hearst, 1992), which are well-motivated by classical artificial intelligence, and turned over to more linguistically motivated tasks such as paraphrase acquisition (Lin and Pantel, 2001a) and linguistic relations such as defined in the ACE tasks (Doddington et al., 2004). With the availability of more and more information captured in text from the web, and with the need to manage this amount of information, knowledge base construction became a new focus of research. TAC KBP aims to be the standardized benchmark to measure progress in this field. As we have outlined in the previous section, the discussed tasks are related yet differ in certain aspects. In the following we will give a roughly chronological overview of the main research literature in areas related to knowledge base population.

Pioneering the field of relation extraction from text on a scale, Hearst (1992) focused on extracting instances of taxonomic `is-a` relations and constructing an ontology from the harvested facts. The basis of this approach are hand-crafted surface patterns (now commonly referred to as *Hearst Patterns* after their inventor) with part-of-speech wildcards. The so-obtained resource is motivated, amongst other things, by linguistic problems and use cases such as synonym recognition and sub-categorization.

Later work has aimed at increasing the number of relations, and at finding measures of relational similarity between surface forms and relations, or even to induce the relations themselves. For measuring the relational similarity between surface forms, two general approaches are possible: First, determining the similarity between two surface forms by statistical qualities of the *argument overlap* linguistic patterns share. Second, taking *structural overlap* of linguistic patterns (such as shared words or parse configurations) as a proxy for semantic similarity.[10]

The DIRT system (Lin and Pantel, 2001a,b) is an early system solving the task of defining similarity by argument overlap in order to induce clusters of paraphrases. Linguistic surface forms are represented as lexicalized dependency paths and grouped together by the so-called *"extended distributional hypothesis: if two paths tend to occur*

[10]It should be noted that these two approaches do not exclude each other: indeed, as will be discussed in later parts of the dissertation, our own top-ranked relation extractors gather training data by argument overlap and generalize from it by structural overlap via features such as skip-n-grams.

in similar contexts, the meanings of the two paths tend to be similar". As a similarity metric, a variant of mutual information is used that takes distributions of paths and arguments into account. The paraphrases are evaluated intrinsically based on similarity ratings, and extrinsically on a question-answering task.

Paraphrase identification can be seen as a pre-cursor to *Open Information Extraction*, which aims at identifying relational clusters, as if they were grouped together intuitively by a human, and representing them in a canonical form. Fader et al. (2011) showed that when no relational schema is given, and syntactic patterns are clustered by their argument overlap, the output often contains many incoherent and uninformative extractions, such as extracting *made("Faust", "a deal")* instead of *made-a-deal-with("Faust", "the devil")*. They propose additional constraints and heuristics to bring open information extraction more into the direction of what a human would intuitively be expecting. Unsupervised open schema relation extraction systems that continuously learn facts from an incoming stream of text are also known under the term of *machine reading* (Etzioni et al., 2006, 2011). Here, the challenges lie also in accommodating new information and in the large scale of data to be handled.

Bollegala et al. (2010) propose a co-clustering algorithm for both surface patterns and argument pairs that is efficient by adding instances sequentially (one-by-one) to the clusters. Nakashole et al. (2012) build a taxonomy of unsupervised relational pattern clusters (called *pattern synsets*) with semantic types. Pattern synsets are based on entity distributions. The patterns are subject to generalization, sequences are partly wildcarded making use of n-gram correlations and Frequent Itemset Mining (Agrawal and Srikant, 1995) to find n-gram combinations with large co-occurrence support. Semantic subsumption statistics are efficiently represented using a suffix tree. Another unsupervised relation clustering algorithm that combines a wide range of different similarity measures, induces argument types and can model polysemy of relational patterns is proposed in Min et al. (2012b).

Relational clustering is related to real-world tasks such as automatic headline generation for news stories: Alfonseca et al. (2013) extract patterns from news headlines and documents within a thematical cluster. Latent variables are learned in a noisy-ormodel. A headline is selected by a two-step random walk from a pattern to latent variables and back to patterns. Their system performs well in an automatic evaluation,

but not with human raters. Balasubramanian et al. (2012) build a graph of unsupervised relations from co-occurrences in text using positive pointwise mutual information as edge-weights and cluster it using Markov clustering (Van Dongen, 2008). The evaluation is done on identifying clusters that correspond to MUC (Chinchor et al., 1993) terrorist events.

For measuring *structural* rather than distributional similarity, structured kernels on syntactic analyses have been applied especially for linguistic ACE-type relations (Zelenko et al., 2003; Bunescu and Mooney, 2005; Mooney and Bunescu, 2005). The early preference for syntactic similarity rather than shallow surface similarities may be partly due to the more interesting structural properties of dependency representations. Also, since ACE relations are more linguistically defined than factual relations as in the KBP paradigm, the usage of linguistic analysis seems to be motivated. However, the observation was early made that shallow features perform better than dependency structures with kernels (Giuliano et al., 2006). Later analysis (Chan and Roth, 2011) showed that even in ACE, 80% of relational mentions are expressed in forms that are not typically assumed to be well-captured by dependency analysis. Although ACE has mainly been superseded by TAC KBP, there is some recent work on the 2005 ACE data that combines tree kernels with LSA and Brown clusters for relation prediction (Plank and Moschitti, 2013).

Downey et al. (2007) establish a ranking function for relational ranking that includes type similarity and elements akin to information retrieval metrics. Although they call their approach *relational language models*, the ranking function is rather related to the Okapi bm25 metric (Robertson et al., 1995) and not to language model approaches in IR (Lafferty and Zhai, 2001).

For supervised relation extraction systems, some guidance has to be provided as to what are desirable extractions. Two approaches are notable here: guiding the system by an existing database, and guiding the system by direct human supervision, that is by providing rules or annotated data. It is evident that in all cases the aim is to keep human effort in guiding the algorithm minimal. Mintz et al. (2009) coin the term *distant supervision* and are the first to use Freebase as the database to generate training data for a knowledge-base population task. The positive training data is obtained by a simple textual match of the information in the knowledge base, special

negative training data is included which is generated from entity pairs that are in none of the considered relations according to the knowledge base. A multi-class logistic classifier is used with lexical and named-entity-tag features, as well as features derived from dependency trees.

Training data that is semi-automatically generated by distant supervision is inherently noisy. In Alfonseca et al. (2012) a generative approach – that was developed originally for multi-document summarization (Haghighi and Vanderwende, 2009) – is used to separate noisy training examples from informative ones. Blessing and Schütze (2012) use various heuristics for matching entities and for mapping between languages to increase the precision of distant supervision.

Minimally supervised algorithms aim at iteratively bootstrapping lexico-syntactic patterns and selectional restrictions from few seed examples per relation, in the extreme case from only one seed pattern (Kozareva and Hovy, 2010a). For bootstrapped learning, the characteristics of the seeds have been found to have a high impact on extraction quality: in (Kozareva and Hovy, 2010b) a model is developed to estimate the usefulness of seeds for bootstrapped learning.

The NELL never-ending language learner (Carlson et al., 2010) is a semi-supervised approach to knowledge base construction. Started in 2010 and seeded with an initial knowledge base, it aims at continuously and automatically incorporating new facts. In order to avoid semantic drift and to maintain a high precision of facts, the system allows for ongoing human feedback and corrections. Other semi-supervised approaches that use spectral embeddings (Bollegala et al., 2011) or lower-dimensional features from topic models (Yao et al., 2011) make use of both unlabeled and labeled training data. A classifier uses the resulting lower-dimensional feature representation, which generalizes better from training data to test data.

Although there are many relations with more than two arguments, most notably event relations (see e.g. Lee et al. (2012)), most of research on relation extraction assumes that relations are binary. It is indeed straightforward to turn n-ary relations into sets of binary relations. However, if those binary relations are not modeled jointly, information may be lost. Bollegala et al. (2013) measure similarity between ternary relations by using kernels on argument features as well as kernels on patterns for clustering them. In a similar vein, the Path Rank Algorithm (PRA) (Lao and Cohen,

2010; Lao et al., 2011) models chains of binary relation to obtain additional evidence for certain facts. PRA finds and ranks inference rules that correspond to random walks in a graph, starting at an entity (the query) and ending at another entity. In this way it can deal with relational information that is expressed by a chain of relations. Different rules (chains of relations) are experts that can be used as features in a log-linear classifier to support other relations. An extension of PRA includes syntactic surface patterns in the model and uses dimensionality reduction for additional smoothness and predictive power (Gardner et al., 2013).

The survey of relevant publications illustrates how research has focused on various interesting problems and aspects in the field of relation extraction and knowledge base population. In order to truly advance the state-of-the-art, it is necessary to identify which of the many possible research questions actually have a major impact on performance in a well defined setting. In the rest of this dissertation, we aim at thoroughly quantifying and improving those places in end-to-end relation extraction that show most potential on this difficult task.

2.5. Summary

In this chapter, we gave an overview of current tasks and approaches in the field of relation extraction. We distinguished linguistic vs. fact-based relation extraction, as well as open and closed schema approaches. We positioned the task of Knowledge Base Population, i.e. extracting and structuring textual information according to specific information needs, in a wide spectrum of challenging problems.

3. Distant Supervision

As we illustrated in Chapter 2, relation extraction can be formulated as the task of turning unstructured text into tabularized information. We distinguished two relation extraction paradigms: 1) open information extraction, the unsupervised clustering of entity-context tuples (Banko et al., 2007), and 2) relation extraction for a fixed relation inventory, which corresponds to the knowledge-base population (KBP) task (Ji and Grishman, 2011). One advantage of open information extraction is that it does not require annotated data. The resulting representation, however, may not always provide the most useful granularity or partitioning for a specific task. In contrast, relation extraction for a pre-specified relation inventory may be better tailored for a specific task, but requires labeled training data; however, textual annotation by hand is costly.

Databases with fact tuples such as *(PERSON, born-in, CITY)* are often readily available. However, there is usually no or only very little text corpora annotated according to whether a relation (e.g. *born-in*) is expressed in a span of text between particular entities (e.g. of types *PERSON* and *CITY*). The paradigm of *distant supervision (DS)* (Craven et al., 1999; Mintz et al., 2009) aims at providing such training data cheaply by using existing knowledge bases: Textual matches of entities from fact tuples are used to automatically generate relation contexts as training instances.

In Section 3.1 of this chapter, we will discuss specific problems that are inherent in semi-automatically generating training data by this method. Besides a qualitative characterization of those problems we will review quantitative characterizations of such problems reported in the literature and give our own estimate of the *false positives* in the data that is used in the context of this work. In Section 3.2 we will characterize existing approaches to dealing with noisy distant supervision training data. We identify three main principles for noise reduction (*at-least-one constraints, generative noise modeling, pattern correlations*) and argue that there exist interesting parallels to redundancy modeling for relation prediction.

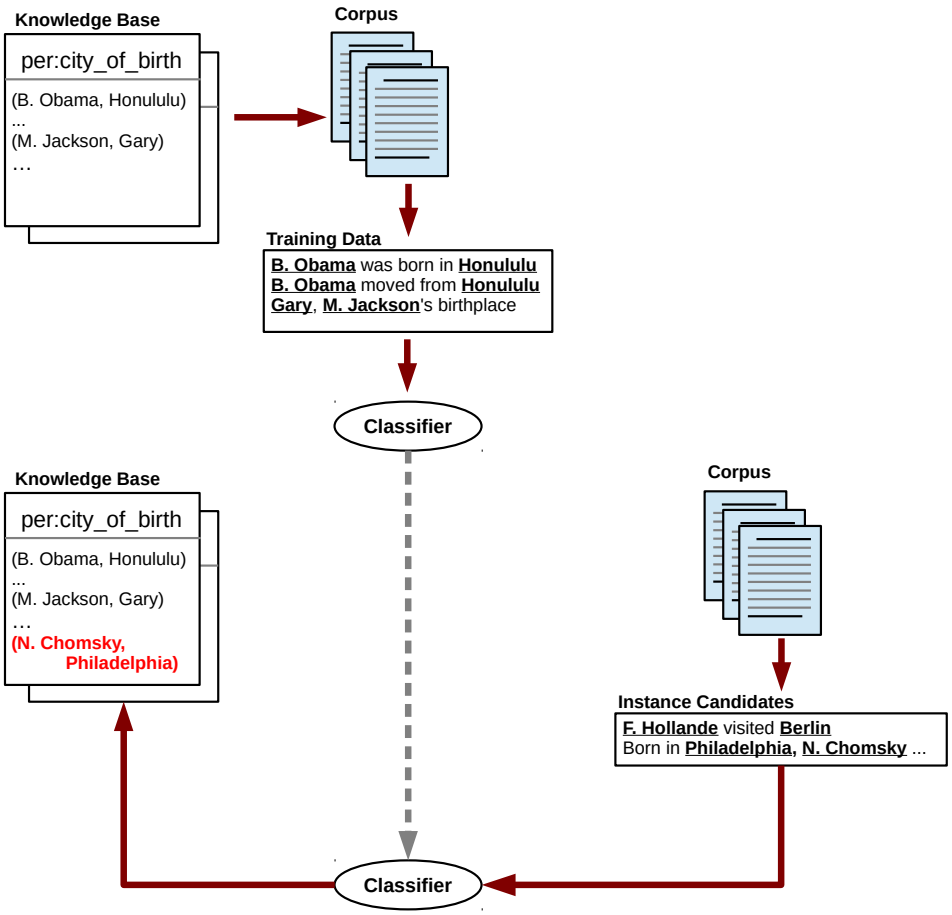


Figure 4: **Distant supervision for knowledge base population. Top, training:** Facts of the knowledge base are matched against a training corpus and noisy instances are obtained for training a classifier. **Bottom, testing:** The resulting classifier predicts relations for candidate sentences of a potentially different corpus, and new facts are added to the knowledge base.

3.1. Problem Statement and Quantification

Figure 4 shows the basic assumed workflow for distant supervision. In a first step, existing extries in a knowledge base are used to find matching argument pairs in a text corpus, and to use the respective contexts as instances for training a relational classifier. In prediction (the second step), the classifier can be used to find new rela-

tional instances from text of the same or another corpus, and populate the knowledge base with the newly extracted information. Often it is only a small fraction of the training matches that indeed express the relation of the fact tuple. For example, the arguments of the fact tuple *("Barack Obama", born-in, "Honululu")* could match in *true positive* contexts like *"Barack Obama was born in Honululu"*, as well as in *false positive* contexts like *"Barack Obama visited Honululu"*.

There are two error cases when extracting training data: *false positive* and *false negative* errors. A *false positive* match is produced if a sentence contains an entity pair for which a relation holds according to the knowledge base, but for which the sentence does not express the relation. The sentence is marked as a positive training example for the relation, however it does not contain a valid signal for it. False positives introduce errors in the training data from which the relational model is to be generalized. For most models *false positive* errors are the most critical error type, for qualitative and quantitative reasons, as will be explained in the following.

A *false negative* error can occur if a sentence and argument pair is marked as a negative training example for a relation (the knowledge base does not contain the argument pair for that relation), but the sentence actually expresses the relation, and the knowledge base was incomplete. This type of error may negatively influence model learning by omitting potentially useful positive examples or by negatively weighting valid signals for a relation. If a (for example generative) model does not include negative training data in learning, or if a model uses the positive instances of other relations as negatives for the relation in question, while false negatives – stemming from missing entries in the KB – influence the amount of training data, the training data are still a representative sample for the relation. For big knowledge bases such as Freebase or Wikipedia infoboxes, the amount of data generated is very big anyway, so missing training data for relations seems acceptable.

Qualitatively, the positive training data is therefore more important and many models do not even make use of explicit negative data but aim to model relations only from positive examples: this is the case for all generative models (see Chapter 3.2.3), but also for the discriminative re-ranker with constraints that we propose in Chapter 4.2. *Quantitatively*, as will be shown below, amongst the data marked *positive* the percentage of errors (coined *false discovery rate* in statistics terminology) tends to be

much higher than the respective percentage of errors in the training data marked as *negative* (*false omission rate*).

For current state-of-the-art system architectures, we generally agree with the focus on improving on *false positive* rather than *false negative* distant supervision errors. At the same time, we point to the theoretical possibility that *false negative* errors could become relevant for systems where the negative training data plays a central role in modeling. In most current systems though, the negative training data comes from simple yet effective heuristics like taking the positive instances from other relations as negative instances for a specific relation.

Work on improving distant supervision errors mostly focuses on identifying the false positives. Often, ranking functions are sought that score contexts expressing the relation higher than contexts matching arguments but not expressing the knowledge base relation. With such a ranking function, training data can be filtered by applying a threshold, or data can be weighted according to the precision estimated by the function. The better the ranking function, the more good data can be retained when filtering.

When the distant supervision errors are quantified, usually the false discovery rate is estimated from manually checking samples of the distant supervision training data. Similarly, the false omission rate can be obtained by sampling directly from the corpus and by checking whether facts are expressed that are not in the knowledge base. In the following we will give an overview of error estimates that can be found in research literature (together with our own false positive estimate). Since error rates vary per relation and the reported estimates are taken from different corpora and sets of relations, they are expected to be different.

The original work on plain distant supervision (Mintz et al., 2009) does not give a quantification of the errors in the training data. Riedel et al. (2010) observe, in their analysis over samples of the relations *nationality, contains* and *place_of_birth*, a false discovery rate (error rate on the positive extractions) of $\frac{FP}{FP+TP} = 31\%$ when aligning Freebase to the New York Times corpus, and a false discovery rate of 13% when aligning to Wikipedia. Hoffmann et al. (2011) sample 100 sentences each for 10 Freebase relations, and get widely varying false discovery rates on the positive extractions ranging from $\frac{FP}{FP+TP} = 11.0\%$ to 99.8% with an average error rate of

Paper	**Setting**	$\frac{FP}{FP+TP}$	$\frac{FN}{FN+TN}$
Riedel	Riedel-Wiki, Riedel	13%, 31%	
Hoffmann	10 rels of Freebase	55.7%	
this work	TAC KBP	37.3%	
Surdeanu	Riedel, TAC KBP	31%, 39%	
Min	Riedel, TAC KBP		8.5%, 11.5%
Xu	50 rels of Freebase	61.4%	5.8%

Table 2: Estimation of error rates for different corpora and relations.

55.7%. Min et al. (2013) provide a statistic according to which the false omission rate (error rate on the negative data), i.e. the contexts with non-matched entity pairs for TAC KBP relations is $\frac{FN}{FN+TN} = 8.5\%$ for the Riedel et al. (2010) dataset, and 11.5% for the TAC KBP dataset (Ji et al., 2010) that was also used by Surdeanu et al. (2012)[11]. The false discovery rates for the Riedel and TAC KBP dataset have been estimated at $\frac{FP}{FP+TP} = 31\%$ and 39% respectively by Surdeanu et al. (2012). Surdeanu et al. (2012) also give an overview of the *overlap* of relations, i.e. the percentage of entity pairs occurring in more than one relation: it is 7.1% in the Riedel data set and relatively low with 2.8% in the TAC KBP data. Xu et al. (2013b) sample 1824 pairs from the the New York Times 2006 corpus and evaluate manually whether they express any of a set of 50 common Freebase relations. Their statistic is summarized as follows: $TN = 90.1$; $FN = 5.5$; $TP = 1.7$; $FP = 2.7$. For our own system, we evaluate the TAC KBP training data obtained from mapped Freebase relations, judge 1243 instances manually and obtain a positive error rate of $\frac{FP}{FP+TP} = 37.2\%$.

Table 2 summarizes these findings. Although the data sets are very different, it is consistently the case that false positive matches degrade the quality of the positive training examples. And while false negatives exist, they do obviously not provide a

[11]This estimate was done on the entity-pair level, the actual FN-rate on the instance level would be lower, because for false negative pairs there are true negative matches to be expected

big fraction of the negative training data; moreover they do not play a central role in the training of most relation extraction models.

3.2. Survey of Noise Reduction Approaches

A number of different approaches have been introduced to automatically determine which training contexts, obtained from relation argument matching, are *true positives*, and which are *false positives*. This chapter aims at giving an overview of approaches tackling this problem (cf. Table 3). They are each based on one of the following principles:

- *At-least-one* constraints state at training time that at least one of the matched contexts for a pair is indeed a true positive – but not necessarily all of them (see Chapter 3.2.1). We deem it potentially fruitful to further research to contrast the at-least-one principle to other schemes applied in prediction (Chapter 3.2.2).
- *Hierarchical topic models* are based on the idea of separating the distributions that generate relation-specific contexts from those that generate pair-specific contexts or background text (Chapter 3.2.3).
- *Pattern correlations* are at the heart of an approach which assumes that training contexts matching argument pairs for a relation either express that relation, or have a large overlap in argument pairs with other patterns expressing the relation. In other words, they explicitly model the fact that a pattern is matching, and exploit this to transfer probability mass to similar patterns (Chapter 3.2.4).

3.2.1. At-least-one Approaches

In the vanilla setting, distant supervision assumes all sentences containing an entity pair to be potential patterns for the relation holding between the entities. As found by Riedel et al. (2010), this assumption quickly becomes untenable when dealing with text data not directly associated with the knowledge base from which the facts are taken. In the following, we describe approaches implementing a relaxing constraint which only presumes that at least one of the entity pair occurrences is a textual manifestation of the relation (*at-least-one* assumption).

Author	**Type**	**Baseline**	**KB** (relations)	**Ground Truth**	**Corpus**
Riedel 2010, Yao 2010	ALO	plain DS, joint model without ALO	Freebase (430)	Freebase, HR	New York Times
Hoffmann 2011	ALO	Riedel 2010	Riedel 2010	Freebase, HR	New York Times
Surdeanu 2012	ALO	plain DS, Riedel 2010, Hoffmann 2011	Riedel 2010, Wikipedia infoboxes (42)	Freebase, TAC key (2010+2011)	New York Times, TAC KBP corpus (2010+2011)
Alfonseca 2012	TM	MLE	Freebase (3)	HR	Web news articles
Takamatsu 2012	PC	plain DS, MLE, MultiR (Hoffmann)	Freebase (24)	Freebase, HR	Wikipedia
Roth (this work)	ALO TM	MLE, Alfonseca 2012	Seeds for TAC relations (42)	TAC key (2009-2011)	TAC KBP corpus (2009-2011)

Table 3: Overview of the experimental settings of the approaches covered in this survey. **Abbreviations: ALO**: at-least-one; **DS**: distant supervision; **HR**: human ratings; **MLE**: maximum likelihood estimate of $P(rel|pattern)$; **PC**: pattern correlations; **TM**: topic model.

Formally, the at-least-one assumption states that

"If two entities participate in a relation, at least one sentence that mentions these two entities might express that relation" (Riedel et al., 2010)

Various models are essentially based on this idea (Riedel et al., 2010; Yao et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012; Roth and Klakow, 2013). Relation classification models are trained with an objective function that includes this constraint. Typically, at-least-one models are multi-class models over a set of relations,

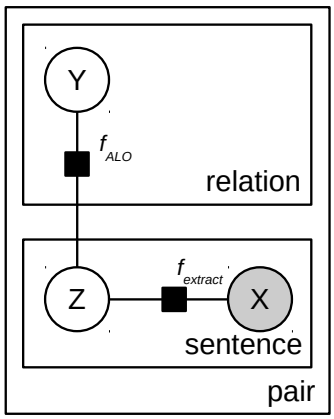


Figure 5: **MultiR in plate notation.** For every entity pair there are multi-class mention variables Z (one for each sentence X observed with the pair) and binary relation variables Y (one for each relation). Each Z can take on one of the relations as a value. A sentence factor scores the compatibility of the relation chosen for Z with the features of the corresponding sentence. The Y variables are relation-specific and indicate whether a particular relation was predicted for the current pair. Mention variables Z and relation variables Y are connected by a dedicated factor that is 1 *iff* the at-least-one assumption is fulfilled.

including a special NIL label to indicate that none of the relations in the knowledge base is expressed by a context.

While the underlying idea regarding noise reduction is the same for all of the at-least-one models, they differ in other assumptions about dependencies in the data, in the point at which the at-least-one constraint is used, and in their inference algorithms. The first proposed model with an at-least-one learner is that of Riedel et al. (2010). It consists of a factor graph that includes binary variables for contexts, and groups contexts together for each entity pair. An entity pair is associated with a variable that can take on a relation value or NIL. A global objective function penalizes the violations of at-least-one constraints, and SampleRank (Rohanimanesh et al., 2011) is used to infer the model.

MultiR (Hoffmann et al., 2011) can be viewed as a multi-label extension of Riedel

et al. (2010). Given an entity pair, the model can predict multiple ("overlapping") relations simultaneously. MultiR models the relation extraction as a factor graph (see Figure 5) with one connected component for each entity pair (e_1, e_2). For each entity pair and relation r, there is a Boolean output variable Y^r to indicate whether the relation is predicted for this entity pair. This set of output variables allows the model to predict multiple (overlapping) relations per pair. The Y variables are distantly supervised in training and they are set to 1 if the respective entity-pair relation tuple is contained in the knowledge base, and set to 0 otherwise. The connected components of the factor graph also model the per-sentence predictions: For each sentence context X_i in which the respective pair occurs, there is one relation picked from the set of relations (including a special label for no relation) $\mathcal{R}$; this per-sentence choice is reflected in the Z_i variables. That is, while several relations can hold for a pair, only one relation can hold per context. The connection between the per-pair variable $Y^r \in \{true, false\}$ and the per-sentence $Z_i \in \mathcal{R}$ variables is established via a dedicated *at-least-one* factor: This factor is 1 if the predictions for r correspond on the pair and sentence level. More formally, the at-least-one factors f_{ALO} connecting the binary per-pair prediction variable for relation r with all per-sentence predictions $\mathbf{Z}$ for the current entity pairs can be written as:[12]

$$f_{ALO}(Y^r, \mathbf{Z}) = \begin{cases} 1 & \text{if } Y^r = true \wedge \exists_i : Z_i = r \\ 0 & \text{otherwise.} \end{cases}$$

The per-sentence predictions are modeled by log-linear factors $f_{extract}$ based on context features ϕ_j modeling the compatibility of sentence X_i with prediction Z_i:

[12] This is the definition of f_{ALO} in the original publication of MultiR (Hoffmann et al., 2011). However, we believe that there is a small error, namely that this formulation would not correctly cover those cases where $Y^r = false$, since in this case the at-least-one objective is fulfilled if there exists no per-sentence prediction for r.

Therefore, the corrected version of the at-least-one factor would be:

$$f_{ALO}(Y^r, \mathbf{Z}) = \begin{cases} 1 & \text{if } Y^r = (\exists_i : Z_i = r) \\ 0 & \text{otherwise.} \end{cases}$$

$$f_{extract}(Z_i, X_i) = exp\left(\sum_j \theta_j \phi_j(Z_i, X_i)\right)$$

The overall model can be stated as

$$P(\mathbf{Y}, \mathbf{Z}|\mathbf{X}; \theta) = \frac{1}{\mathcal{Z}_\mathbf{X}} \prod_r f_{ALO}(Y^r, \mathbf{Z}) \prod_i f_{extract}(Z_i, X_i)$$

where $\mathcal{Z}_\mathbf{X}$ is the normalization constant. A perceptron training scheme is employed to estimate the parameters.

A further extension is MIMLRE (Surdeanu et al., 2012), a jointly trained two-stage classification model. MIMLRE, on one layer, makes multi-class predictions for contexts. The predictions of this layer are used by a collection of binary per-relation classifiers to predict the labels for an entity pair. Instead of hard-coding the at-least-one requirement (as in MultiR), in MIMLRE the per-pair aggregation of per-sentence predictions is another classification task based on aggregate features of the per-sentence predictions. That is, while the basic topology of the MIMLRE model is the same as in MultiR, the main difference is that the binary at-least-one logic (modeled in MultiR by f_{ALO}) is replaced by a feature-based predictor modeling $P(Y|\mathbf{Z}; \theta)$. The at-least-one semantics is brought into the model by a special feature in the per-relation classifiers, indicating whether the relation in question r was predicted at least once in $\mathbf{Z}$. For each relation $r' \in \mathcal{R} \setminus r$, additionally a joint feature is instantiated if r and r' where both predicted at least once in $\mathbf{Z}$.

Most at-least-one approaches require dedicated negative training data to estimate enough probability mass for the NIL class. For MultiR, Hoffmann et al. (2011) use 10% of the pairs occurring in the text, but not in the knowledge base, as negative training data. For MIMLRE, Surdeanu et al. (2012) use between 5% and 10% of subsampled negative examples, depending on the data set. We will propose an at-least-one model that does not require negative training data while enforcing additional ranking constraints for NIL in Chapter 4.2.

It is interesting to note that, while the term distant supervision was coined by Mintz et al. (2009), and at-least-one learning for distant supervision was introduced by Riedel et al. (2010), learning relational extractors from seed pairs (albeit only a handful and not from a knowledge base) had already been considered by Bunescu and

Mooney (2007). In their experiments, they already mention the possibility of at-least-one learning (also referred to as *multiple instance learning, MIL*), however they only approximate it in their experiments by changing the cost function for false negatives in a standard support vector machine, as proposed in Ray and Craven (2005).

3.2.2. Connection to Predictive Redundancy Models

Many relation extraction systems (Kasneci et al., 2009; Roth et al., 2013) decide whether a fact is extracted or not at prediction time according to the following simple rule: A fact is extracted if and only if there is a positive decision for at least one context. For context occurrences $c_i, i = 1...n$ of an argument pair a_1, a_2, this decision rule decides whether the fact $r(a_1, a_2)$ for a relation r holds, by the scoring formula:

$$P(r|a_1, a_2) = \max_i P(r|c_i, a_1, a_2)$$

This prediction rule has its corresponding counterpart on the training side with at-least-one-context training.

A straightforward continuous generalization of this rule is to assign a score by noisy-or (Lin et al., 2003). Noisy-or prediction corresponds to using the scoring function:

$$P(r|a_1, a_2) = 1 - \prod_i (1 - P(r|c_i, a_1, a_2))$$

The noisy-or formula obviously is a smoother measure. However, it too is strongly influenced by high-scored patterns: the fact probability is *at least* as big as the maximum context probability for that pair.

At-least-one-context and noisy-or schemes are simple examples of redundancy models, i.e. models that combine scores for several instances to an overall prediction. Explicitly modeling the step from scoring a context to predicting whether a fact holds or not is a technical requirement at prediction time, for every relation extraction system. At training time, technically this step can be circumvented by only estimating context predictors and, in the case of distant supervision, ignoring that training data is noisy. It is obvious, though, that such an approach does not give optimal results. Because of this asymmetry, more research was done in the past on redundancy modeling for predicting relations (rather than for the training step), and terminology and

$n = \frac{\text{sentences}}{\text{pair}}$	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8
% data	8.3	7.1	8.0	6.6	8.8	10.1	10.5	11.1	29.5
at-least-k	0	0	1	3	7	16	34	72	149

Table 4: **Distribution of training data and 95%-quantile for observing at least n sentences.** Pairs from the knowledge base (Freebase) are grouped into buckets by the number of sentences in the TAC KBP source corpus containing them. Buckets contain pairs matching n up to the next higher number in the table sentences. % *data* shows how much of the training data lies in the resulting buckets. *at-least-k* is the estimate at least how many k positive sentences to expect per pair in each bucket, with at-least 95% confidence, assuming a constant *true positive*-rate of $p = 63\%$.

usage are sometimes inconsistent between training and prediction. MIMLRE (Surdeanu et al., 2012) for example, uses an at-least-one-context scheme for training, but noisy-or for prediction. While at-least-one-context models have been extensively studied for training – equivalent to at-least-one prediction – less work has been done on noisy-or training (Takamatsu et al. (2012), however, use noisy-or in their correlation calculation).

Both views (at-least-one and noisy-or) do not explicitly take into account the number of contexts for a fact triple scored low by the model. Instead, such objective functions tend to be influenced mainly by the contexts (for each candidate triple) that are scored high by the model. Since the overall number of contexts for a candidate tuple is not included in the model, large numbers of contexts that are given a low probability for the relation do not influence the score negatively. This has been identified as a problem for prediction by Downey et al. (2005), and led to the development of the probabilistic URNS model which expects particular minimal ratios of *true* and *false* contexts, depending on the number of contexts for a fact. We assume that similar models might be beneficial during training by relaxing the at-least-one constraint for singleton tuples and requiring more positive instances for frequently matching tuples.

We illustrate this point by the following estimation (see Table 4): We randomly sampled 1.243 sentences out of those sentences that contained argument entity pairs

of Freebase relations in the TAC KBP corpus. The Freebase relations were chosen to correspond to the 42 TAC KBP relations. The selected sentences were manually annotated, and it was marked whether they indeed expressed the respective relation or not. We observed a *true positive*-rate of $\frac{TP}{TP+FP} = 63\%$. The annotated matches were grouped according to the total number of matches of the corresponding pairs[13]. The *true positive*-rate of 63% is within the standard error, i.e. roughly the same, for pairs with both low and high numbers of matches. To answer the question *"At least how many true positives can one expect with a confidence of* 95%*?"*, we calculate the 95%-quantile (inverse cumulative distribution function) of $P(X \geq k)$ for $X \sim B(n, p)$, distributed according to a binomial distribution, where p is the observed *true positive*-rate and n the number of matching sentences per pair. (We make the simplifying assumption of using the same true positive rate across all relations, as per-relation estimates would be too unreliable.) Table 4 shows the selected values for n with the respective value of at-least-k.[14] We also report the amount of training data in bins up to the next shown value for n. One can see that at-least-1 is too strong an assumption for roughly 15% of the data, while for the big majority of training data much stronger

[13]The training data has a cut-off of 500 sentences per pair.

[14]A note on the calculation of these statistics. Math packages usually have a special function to compute quantiles for the Bernoulli cumulation distribution function (cdf) $P(X \leq k)$, $X \sim B(n, p)$. However, we are interested in the quantile of $P(X \geq k) = 1 - P(X \leq k-1)$. We use the definition of the cdf via the incomplete beta function $I.(\cdot, \cdot)$, and define a second cdf $\bar{P}(X \leq k)$, $X \sim B(n, 1-p)$:

$$P(X \geq k) = I_{1-p}(n-k, k+1)$$

$$= 1 - I_p(k+1, n-k)$$

$$= 1 - \bar{P}(X \leq n-k-1)$$

where the first and third steps result from the definition of the cdf via the incomplete beta function, and the second from the properties of the incomplete beta function. Now, the original cdf can be re-written as:

$$P(X \geq k) = 1 - P(X \leq k-1)$$

$$= 1 - (1 - \bar{P}(X \leq n - (k-1) - 1))$$

$$= \bar{P}(X \leq n-k)$$

replacing $\bar{k} = n - k$ and calculating the 95% quantile of $\bar{P}(X \leq \bar{k})$ gives us the result $\bar{k}$, and by solving for $k = n - \bar{k}$ we have the desired quantity.

assumptions can be made.

To summarize, the *at-least-one* assumption that builds the basis of many approaches to noise reduction (including the one proposed in Chapter 4.2) is only one of several possibilities to use a redundancy model for training with distant supervision data. Redundancy models with more connections to probability theory – such as noisy-or or URNS – are more complex and would be more difficult to incorporate into predictive models. Research on distant supervision models in the redundancy elimination category has therefore focussed on the simpler *at-least-one* principle. It remains an open question whether more sophisticated redundancy models can profit distantly supervised relation extraction.

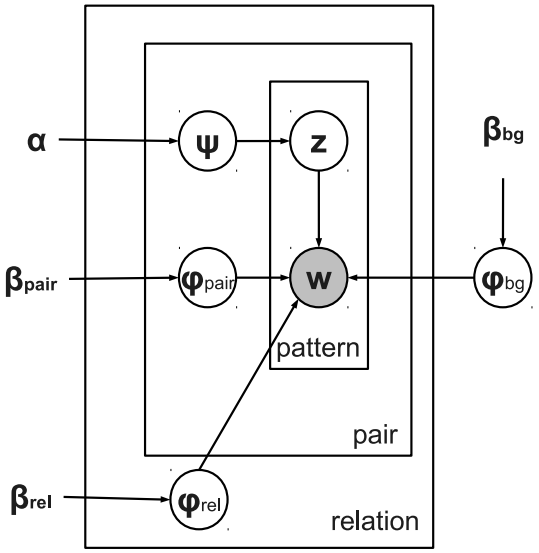


Figure 6: **Hierarchical topic model for distant supervision.** Context patterns are generated by either a background, relation-specific or pair-specific distribution.

3.2.3. Hierarchical Topic Models

As was illustrated in the previous chapter, at-least-one schemes allow a relation prediction model some flexibility to account for noise on a per pair level. A discriminative classifier meets the overall objective if at least the minimally expected number of sentences per pair is classified as *true*. In this chapter we will describe another approach to separating actual from spurious relational matches by generatively separating relation and noise distributions.

The hierarchical topic model (*HierTopics*) introduced by Alfonseca et al. (2012) is a generative model to score and filter relational context patterns. It assumes that a context pattern matching an entity pair in the knowledge base for a particular relation is either typical for the entity pair, for the relation, or for neither of the two. This principle is then used to infer distributions of one of the following types:

1. For every entity pair, a pair-specific distribution (over patterns).
2. For every relation, a relation-specific distribution.
3. A general background distribution.

It is inspired by the hierarchical topic model for multi-document summarization of Haghighi and Vanderwende (2009): One can view the surface patterns as words, and the argument pairs as the documents that contain those words. Additionally to the model of Haghighi and Vanderwende (2009), the entity pairs ($\sim$ Haghighi: *documents*) are grouped together on yet another level, namely according to the relation they stand in.

The generative process assumes that for each argument pair of a particular relation, all patterns (i.e. surface strings or dependency paths between arguments from distant supervision matches) are generated by first choosing a hidden variable z at a position i, depending on a pair-specific distribution ψ (with Dirichlet hyper parameters α). The variable z can take on three values, B for background, R for relation and P for pair. Corresponding vocabulary distributions ($\phi_{bg}, \phi_{rel}, \phi_{pair}$) are chosen to generate the context pattern at position i. The vocabulary distributions are smoothed by Dirichlet hyper parameters $\beta_{bg}, \beta_{rel}, \beta_{pair}$ and shared on the respective levels. See Figure 6 for a plate diagram of the HierTopics model. Gibbs sampling is used to infer the topics of the document collection.

The HierTopics model aims at separating out the relation vocabulary in an efficient and elegant way. Compared to at-least-one models it allows for a desired degree of freedom: The amount of positively labeled contexts per entity pair is dependent on the vocabulary and not on fixed ratios or *at-least-k* numbers, and no statistics about *true* and *false positives* need to be gathered. Moreover, with such a generative model no dedicated negative training data is required. On the other hand, at-least-one learning has been demonstrated to be an apt building block (e.g. as a dedicated node in a factor graph) in more complex models using effective discriminative training schemes.

The original hierarchical topic model (Alfonseca et al., 2012) treats patterns as a whole. In our work (Chapter 4.1.2) the model is extended by a second layer of hidden variables in order to include bi-gram features for improving estimates for the long tail of infrequent patterns for which evidence on pattern level alone may be too sparse. The comparison to an at-least-one perceptron learner (see Chapter 4.5) shows that while the simple surface-pattern topic model version is better than a baseline using relative frequency counts, it is not as good as the at-least-one model. By including features in the hierarchical topic model its performance comes very close to that of the

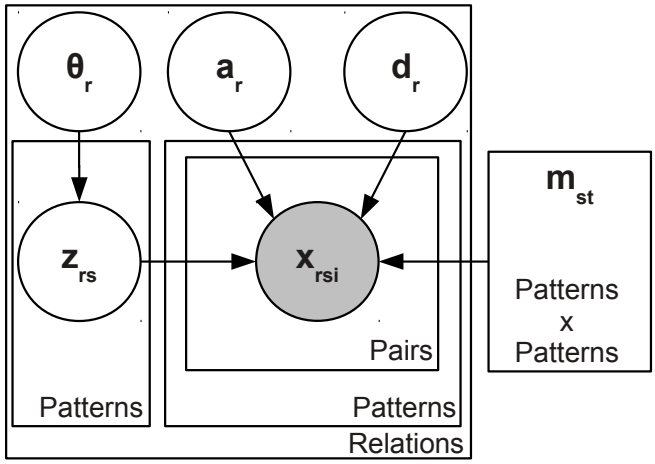


Figure 7: **Plate diagram for the Takamatsu model.** The hidden variables z_{rs} indicate whether a relation r is expressed by a pattern s. The observed variables x_{rsi} denote which contexts are matched by an argument pair i. θ_r, a_r and d_r are the parameters to be learned, m contains the correlation statistics.

perceptron.

3.2.4. Pattern Correlations

While the *HierTopics* approach, described in the previous section, models the distant supervision corpus by a generative process and then obtains information about relevance of patterns as a by-product, Takamatsu et al. (2012) aim more directly at modeling the question whether a pattern expresses a relation or not. The underlying idea is that context patterns which match argument-pairs for a relation either express that relation, or have a high overlap in argument pairs with other patterns expressing the relation (or, none of the two, which case is covered by an additional constant probability). The argument pairs of patterns that express a relation may still frequently co-occur with other patterns that do not express the relation.

To give an example, given some patterns *s="[ARG1] and [ARG2]"* and *t="[ARG1] is the wife of [ARG2]"*, if there is a context

"[Michelle Obama] and [Barack Obama]" = s([MO], [BO])

the context (or, rather its pattern s) can be labeled negative for a relation *spouse_of* if pattern t is labeled positive and $P(pair \in s|pair \in t)$ is high. Note that it is not necessary that the actual context

"[Michelle Obama] is the wife of [Barack Obama]" = t([MO], [BO])

is present in the training data. This is a major difference to at-least-one training schemes. To give a different example, a *negative* label for the same statement $t([MO], [BO])$ could *not* be explained by a positive label for *"[ARG1] and [ARG2]"* if $P(pair \in t|pair \in s)$ is small – pattern t would have to be assigned a positive label to fulfil the model requirements. The pattern co-occurrence probabilities are calculated prior to inference based on the overlap of sets of entity pairs matched by the patterns.

A probabilistic graphical model (see Figure 7) is learned that contains hidden variables z_{rs} indicating whether a pattern s indeed expresses a relation r. The topology of the model is different from *HierTopics*: Although the observed variables are tuples of patterns and argument pairs in both cases, Takamatsu et al. group the contexts by patterns and do not consider repeated occurrences of contexts.

The rationale behind the probabilistic process is the following: If a tuple of a relational pattern s and argument pair i is observed, and argument pair i is in the knowledge base, then this can have one of the following causes:

1. Pattern s expresses relation r, i.e. $z_{rs} = true$.
2. Pattern s does not express relation r – however, some other pattern t expresses r and arguments of t are often arguments of s, i.e. $z_{rt} = true$ and $P(pair \in s|pair \in t)$ is high.
3. Pattern s does not express relation r – however, the existence of fact i in the knowledge base is explained by some other process not captured by the model.

That is, the model deals separately with case 1, when the underlying variable for the pattern directly expresses the fact in the knowledge base (relation r holds for the argument pair), and cases 2 and 3, when the argument pair is in the knowledge base but the pattern does not express r. The model estimates parameters for case one and a probability for case three, and infers the hidden variables z_{rs}. The probabilities for case two can be obtained from the data prior to training. For case two, it is not

necessary that another pattern occurs with argument pair i, as it would be the case in an at-least-one setting. In this way, the model can hypothesize whether an entity pair i could have been generated by another pattern t expressing r, even if t and i have never been observed together in the corpus.

3.3. Summary

In this chapter, we outlined the semi-supervised generation of training data by *distant supervision* and characterized the types of errors contained in such data both in qualitative and quantitative terms. We further grouped existing approaches for noise reduction into three categories (at-least-one, topic models, pattern correlations), based on the principle they employ to give the model the flexibility to ignore or downweigh noisy argument matches in the training data. In the next chapter, we build on two of these principles: we combine the principle of at-least-one constraints with constraints on the *ranking* of instances, and we extend a generative topic model to include features instead of operating on context patterns only.

4. Better Noise Reduction: A Feature-Based Topic Model and a Novel At-Least-One Ranker

In this chapter we propose two extensions to discriminative and generative distant supervision modeling and combine the output of a discriminative at-least-one learner with that of a generative hierarchical topic model to reduce the noise in distant supervision data. The combination increases the ranking quality of extracted facts and achieves state-of-the-art extraction performance in an end-to-end setting.

As mentioned in the previous chapter, three basic approaches have been proposed to deal with noisy distant supervision instances: The *discriminative at-least-one* approach (Riedel et al., 2010), that requires that at least one of the matches for a relation-entity tuple indeed expresses the relation; The *generative* approach (Alfonseca et al., 2012) that separates relation-specific distributions from noise distributions by using hierarchical topic models; And the *pattern correlation* approach (Takamatsu et al., 2012) that assumes that patterns which match argument pairs have a large overlap in argument pairs with other patterns expressing the relation.

In the following we introduce and combine 1) a *discriminative at-least-one* learner, that requires high scores for both a dedicated noise label and the matched relation, and 2) a *generative topic model* that uses a feature-based representation to separate relation-specific patterns from background or pair-specific noise. The *discriminative* model is novel in that the noise class (represented by a *NIL* label) is enforced by a *ranking constraint* that allows for learning a perceptron model without specifying explicit negative training data. Therefore, one of the the advantages of generative modeling is brought into a discriminative setting. The extended *generative* model is novel in that it incorporates features, which was previously only done for discriminative models (both Takamatsu et al. (2012) and Alfonseca et al. (2012) operate on the pattern level).

We score relational contexts and show that combining the two approaches results in a better ranking quality of relational facts. In an end-to-end evaluation we set a threshold on context pattern scores and apply the patterns in a TAC KBP-style evaluation. Although the finally applied surface patterns are very simple, they achieve state-of-the-art extraction results.

4.1. Hierarchical Topic Model

In Chapter 3.2.3 we described the hierarchical topic model of Alfonseca et al. (2012) which uses contextual patterns as the basic building blocks. It assumes that a context pattern matching an entity pair in the knowledge base for a particular relation is either typical for the entity pair, the relation, or neither. Patterns frequently matching the fact tuple `spouse(Michelle Obama, Barack Obama)`, would include for example:

1. *[ARG1] and president [ARG2]*
2. *[ARG2] 's wife [ARG1]*
3. *[ARG1] with [ARG2]*

Here, intuitively, the first context would belong to the *entity pair*-category, the second to the category expressing the *relation*, and the third would be categorized as a generally frequent background pattern.

4.1.1. Original Model

The generative process assumes that for each argument pair of a particular relation, all patterns (surface strings between arguments from DS matches) are generated by first choosing a hidden variable Z at a position i, depending on a pair-specific distribution ψ (with Dirichlet hyper parameters α). The variable Z can take on three values: B for background, R for relation and P for pair. Corresponding vocabulary distributions $(\phi_{bg}, \phi_{rel}, \phi_{pair})$ are chosen to generate the context pattern W at position i. The vocabulary distributions are smoothed by Dirichlet hyper parameters $\beta_{bg}, \beta_{rel}, \beta_{pair}$ and shared on the respective levels. See Figure 6 on page 33 for a plate diagram of the basic HierTopics model.

In our experiments we use Gibbs sampling (Griffiths and Steyvers, 2004) to infer the topics of patterns. Topics are sampled based on the equations below, where we make use of the following notations: $n_{-i}(\dots)$ refers to counts excluding the current position i; $pair(i)$ refers to the argument entity pair from which the pattern at position i originates; $rel(i)$ refers to the corresponding relation.

$$P(W_i = w | Z_i = P) = \frac{n_{-i}(w, P, pair(i)) + \beta_{pair}}{n_{-i}(P, pair(i)) + W\beta_{pair}}$$

$$P(W_i = w|Z_i = R) = \frac{n_{-i}(w, R, rel(i)) + \beta_{rel}}{n_{-i}(R, rel(i)) + W\beta_{rel}}$$

$$P(W_i = w|Z_i = B) = \frac{n_{-i}(w, B) + \beta_{bg}}{n_{-i}(B) + W\beta_{bg}}$$

$$P(Z_i = z|pair(i)) = \frac{n_{-i}(z, pair(i)) + \alpha_z}{n_{-i}(pair(i)) + \sum_z \alpha_z}$$

A topic z is then sampled proportionally to the following product:

$$P(Z_i = z|pair(i))P(W_i = w|Z_i = z)$$

These sampling equations also build the basis for our feature-based extension to that model described in the following, which is the first generative noise reduction model that incorporates feature-based representations.

4.1.2. Extended Model: Hierarchical Topic Model with Features

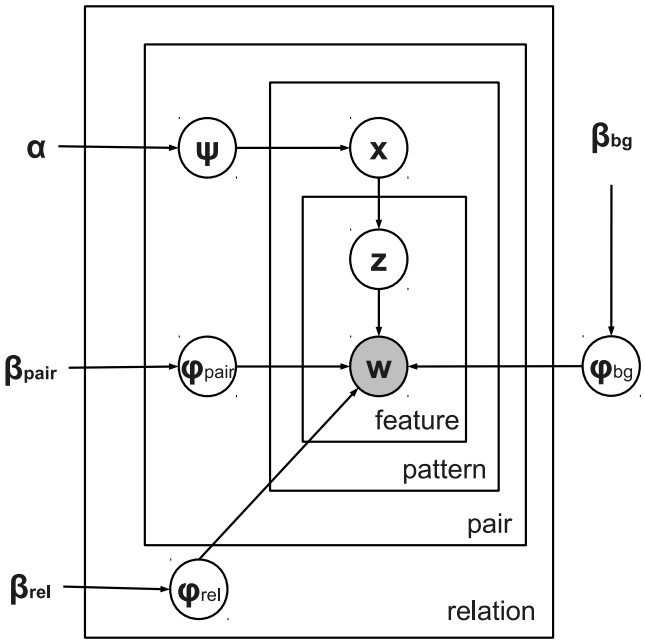


Figure 8: Feature-based hierarchical topic model.

The intertext topic model of Alfonseca et al. (2012) can only treat patterns as a whole. We extend the model to include bi-grams for generalizing over patterns. Intuitively, taking again the fact tuple `spouse(Michelle Obama, Barack Obama)` as the

running example, certain bi-grams would be indicative for one of the three categories for a distant supervision match:

1. *"president [ARG2]"*, for pair-specific contexts, such as *"President Barack Obama, center, is flanked by Michelle Obama"*.

2. *"wife [ARG1]"*, for contexts of those pairs for which the relation holds, such as *"Frank Sinatra and his first wife Nancy Barbato"*.

3. *"with [ARG2]"*, for contexts frequently occurring with pairs from any relation.

In order to include features in the model, we employ a model with two layers of hidden variables. A variable x represents a choice of B, R or P for every pattern. The generated observations in this model are not the patterns (as in the simple intertext model) though, but features W. Each feature W_i is generated conditioned on a second variable $z \in \{B, R, P\}$. The index that the features range over is denoted by i, patterns range over an index j. For a pattern at index j, first one hidden variable x is generated, then all z variables are generated for the corresponding features at indices i (see Figure 8). The values B, R or P of z depend on the corresponding x by a transition distribution:

$$P(Z_i = z | X_{j(i)} = x) = \begin{cases} p_{same}, & \text{if } z = x \\ \frac{1-p_{same}}{2}, & \text{otherwise} \end{cases}$$

where a function $j(i)$ is used to denote the mapping from a feature index i to the index j of the corresponding pattern; p_{same} is set to .99 to enforce the correspondence between pattern and feature topics.[15]

4.2. At-least-one with NilBoost

As a second feature-based model, we propose a perceptron model with an objective function that enforces certain constraints. The model includes log-linear factors for the set of relations $\mathcal{R}$ as well as a factor for the *NIL* label (no relation). Probabilities for

[15] While the original work reports hyper parameters $\alpha = (15, 1, 15)$, we found a uniform prior $\alpha = (1, 1, 1)$ to work slightly better, which we use for the feature-based experiments.

Algorithm 1 At-Least-One Perceptron Training with NilBoost

$\theta \leftarrow 0$
for $r \in \mathcal{R}$ **do**
  for $pair \in$ kb_pairs(r) **do**
    for $pat \in$ sentences$(pair)$ **do**
      for $r' \in \mathcal{R} \setminus r$ **do**
        if $P(r|pat, \theta) \leq P(r'|pat, \theta)$ **then**
          $\theta \leftarrow \theta + \phi(pat, r) - \phi(pat, r')$
        if $P(NIL|pat, \theta) \leq P(r'|pat, \theta)$ **then**
          $\theta \leftarrow \theta + \phi(pat, NIL) - \phi(pat, r')$
    if $\forall_{pat \in sentences(pair)} : P(r|pat, \theta) \leq P(NIL|pat, \theta)$ **then**
      $pat^* = \arg\max_{pat} \frac{P(r|pat,\theta)}{P(NIL|pat,\theta)}$
      $\theta \leftarrow \theta + \phi(pat^*, r) - \phi(pat^*, NIL)$

a relation r given a sentence pattern *pat* are calculated by normalizing over log-linear factors:

$$P(r|pat, \theta) = \frac{f_r(pat)}{\sum_{r' \in \mathcal{R} \cup NIL} f_{r'}(pat)}$$

The factors are defined as:

$$f_r(pat) = exp\left(\sum_i \phi_i(pat, r)\theta_i\right)$$

with $\phi(pat, r)$ the feature vector for sentence pattern *pat* and label assignment r, and θ_r the feature weight vector. Since the decision is whether the distance supervision training example expresses the relation in question r or is noise (*NIL*), the feature-based part of the pattern scoring function is the following ratio:

$$\frac{P(r|pat, \theta)}{P(r|pat, \theta) + P(NIL|pat, \theta)}$$

The learner is directed by the following semantics: First, for a DS sentence with a pattern *pat* matching two arguments for relation r, relation r should have a higher probability than any other relation $r' \in \mathcal{R} \setminus r$. This constraint incorporates the distant supervision signal from the knowledge base. Second, as extractions are noisy, we also expect many contexts to have a high probability for NIL. We therefore introduce the

constraint that NIL has a higher probability than any relation $r' \in \mathcal{R} \setminus r$. While the two top-ranked labels are bound to be either r or NIL, the model has some freedom as to which of the two receives the higher score given the features of the context. Third, at least one DS sentence for an argument pair is expected to express the corresponding relation r. This is required for the model to attribute model capacity to actual relation modeling (rather than assigning the NIL label to all contexts). For patterns pat_i from sentences that contain an entity pair belonging to relation r, this can be written as the following constraints:

$$\forall_{i,r'} : P(r|pat_i) > P(r'|pat_i) \wedge P(NIL|pat_i) > P(r'|pat_i)$$

$$\exists_i : P(r|pat_i) > P(NIL|pat_i)$$

Hence, the first constraint ensures that one of the acceptable labels r or NIL is predicted, while both acceptable labels are constrained to be top-ranked. The at-least-one assumption is ensured by the second constraint. The violation of any of the above constraints triggers a perceptron update. The update corresponding to a violated *at-least-one* constraint is applied only to the one sentence that already has the highest score for the correct label.

The training procedure is outlined in Algorithm 1: The feature vector is initiated in line 1. All entries in the knowledge base are iterated over (see lines 2 and 3). For each argument pair in the knowledge base, the matching sentences from the text corpus, and the corresponding patterns connecting the relational arguments are retrieved (line 4). The ranking constraints are checked in lines 5 to 9: If some relation r' is ranked higher than the relation expressed in the knowledge base, a gradient update (Collins, 2002) to feature vector is performed (line 7). Likewise, the feature vector is updated if any relation not supported by the knowledge base is ranked higher than the NIL label (line 9). The feature vector is updated in line 12 if the at-least-one constraint is violated. Here the update pertains to the features of that sentence which already had the highest score for the sought relation. The actual implementation additionally uses averaging over all updates, and lets the algorithm iterate 20 training passes over the data.

We also experimented with changing the at-least-one instance constraint to requiring at least $n\%$ of the training data, with varying n; this, however did not improve the

performance. To conclude, this discriminative algorithm does not require negative training data nor ratios of negatives.

4.3. Model Combination

The per-pattern probabilities $P(r|pat)$ are calculated for each of the methods and aggregated over all pattern occurrences: For the topic model, the number of times the relation-specific topic has been sampled for a pattern, $n(pat, topic(r))$, is divided by $n(pat)$, the number of times the same pattern has been observed. To be unambiguous, $n(pat, topic(r))$ is the number of occurrences of a context pattern pat with training entity pairs for relation r, when the topic model has sampled $Z_i = R$ for the underlying hidden variable. Analogously for the perceptron, the number of times a pattern co-occurs with entity pairs for r is multiplied by the raw perceptron score $\frac{P(r|pat,\theta)}{P(r|pat,\theta)+P(NIL|pat,\theta)}$ and divided by $n(pat)$. That is, for the perceptron method each pattern occurrence is weighted by its relation score, and all weighted patterns are aggregated.[16] To summarize, for the patterns of the form ***[ARG1]*** *context* ***[ARG2]***, we compute the following scores:

- **Maximum Likelihood (MLE)**:

$$\frac{n(pat, r)}{n(pat)}$$

- **Topic model**:

$$\frac{n(pat, topic(r))}{n(pat)}$$

- **Perceptron**:

$$\frac{n(pat, r)}{n(pat)} * \frac{P(r|pat, \theta)}{P(r|pat, \theta) + P(NIL|pat, \theta)}$$

The topic model and the perceptron approaches are based on plausible yet fundamentally different principles of modeling noise without direct supervision. It is therefore an interesting question how complementary the models are and how much can be gained from a combination. As the two models do not use direct supervision, we also avoid tuning parameters for their combination.

[16] This combination is beneficial since that way the model can make use of both the surface form as well as the relative frequency.

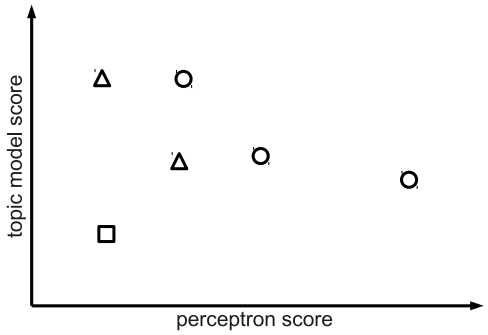


Figure 9: **Score combination by non-dominated sorting.** Circles indicate patterns on the Pareto-frontier, which are ranked highest. They are followed by the triangles, the square indicates the lowest ranked pattern in this example.

We use two schemes to obtain a combined ranking from the two model scores: The first is a ranking based on non-dominated sorting by successively computing the Pareto-frontier of the 2-dimensional score vectors (Borzsony et al., 2001; Godfrey et al., 2007). The underlying principle is that all data points (patterns in our case) that are not dominated by another point[17] build the frontier and are ranked highest (see Figure 9), with ties broken by linear combination. Sorting by computing the Pareto-frontier has been applied to training machine translation systems (Duh et al., 2012) to combine the translation quality metrics BLEU, RIBES and NTER, each of which is based on different principles. In the context of machine translation it has been found to outperform a linear interpolation of the metrics and to be more stable to non-smooth metrics and non-comparable scalings. As a second combination scheme, we include a simple linear interpolation with uniform weights in our comparison.

4.4. Universal Schema

In order to compare the noise reduction models to a state-of-the-art method for relation prediction that follows an entirely different paradigm, we include *Universal Schema*, which is based on matrix factorization (Riedel et al., 2013), in our ranking experiments. The idea behind Universal Schema is to achieve generalization by constraining the

[17] A data point h_1 dominates a data point h_2 if $h_1 \geq h_2$ in all metrics and $h_1 > h_2$ in at least one metric.

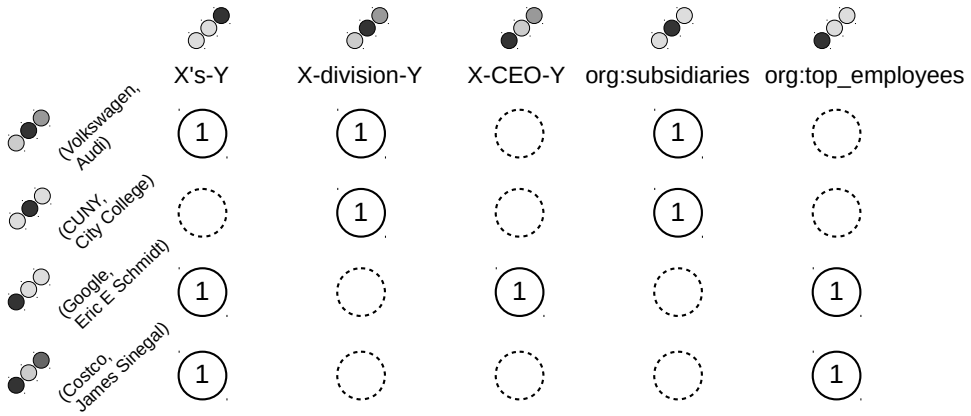


Figure 10: **Universal Schema matrix for distant supervision training data.** The matrix contains entries for observed co-occurences of argument pairs with textual context patterns as well as for database entries (argument pairs with relations). The model estimates vectors for rows and columns, so that the similarity for these vectors is higher for observed cells than for unobserved ones.

model to be expressed with a lower-dimensional representation. By this way the model should be able to make use of indirect correlations, and to make meaningful predictions even for pairs that do not have many surface patterns co-occurring.

Universal Schema had originally been developed in the context of recommender systems and collaborative filtering (Rendle et al., 2009). The model starts with a training matrix Y, the rows of which correspond to entity *tuples* $t \in T$, and the columns of which correspond to *relations* $r \in \mathcal{R} \cup \mathcal{S}$, where $\mathcal{R}$ contains the relations to be predicted, and $\mathcal{S}$ contains the surface patterns (which are given the same status as relations in the model, see Figure 10). Each non-zero cell captures the co-occurrence of a relation (or pattern) r with a tuple t, observed in the distant supervision KB or from a distant supervision textual match. The model approximates the matrix using a natural parameter $\theta_{r,t}$ and the logistic function:

$$p(y_{r,t} = 1|\theta_{r,t}) := \sigma(\theta_{r,t}) = \frac{1}{1 + exp(-\theta_{r,t})}$$

The parameters $\theta_{r,t}$ are defined through latent feature vectors associated with the

rows (tuples) and columns (relations). Given K-dimensional latent feature vectors $\mathbf{a_r}$ and $\mathbf{v_t}$ for a relation r and tuple t the natural parameter is obtained by:

$$\theta_{r,t} := \sum_{k=1}^{K} a_{r,k} v_{t,k}$$

The resulting vectors $\mathbf{a_r}$ and $\mathbf{v_t}$ of Universal Schema correspond to generalized PCA (Collins et al., 2001). Instead of directly optimizing the likelihood of the matrix *cells* Y, we use *Bayesian Personalized Ranking (BPR)* (Rendle et al., 2009) (as also used in Riedel et al. (2013)) to use stochastic gradient descent for optimizing the likelihood of the *pairwise ranking* of observered over unobserved cells per column. In our experiments we perform 1000 training epochs, and set the regularizer for component weights to 0.1. [18]

Optimizing the vector representations for the pairwise ranking with BPR bears certain characteristics of noise reduction techniques: If a value in the matrix is not observed (e.g. an argument-relation-tuple was not in the training knowledge base), the only requirement in training is that this pair should have a lower θ-value than a pair that was observed with this relation - not a low θ-value in absolute terms. This can intuititvely be motivated for the distant supervision setting by the fact that we are confident in the signals from the data-base, and also want to allow for maximal flexibility in scoring argument-relation-tuples for which we do not have explicite training signals. Additionally to the BPR experiments (*USchemaBPR*), we also include a run *USchemaDirect*, which employs direct optimization of the cell values, with negative cells randomly sampled from non-observed cells.

We use the Universal Schema vectors to obtain the cosine similarity between the vectors of TAC KBP relations and the pattern vectors. These scored patterns are then used to match and predict answer from the candidate set. This corresponds to the method for provenance finding of extracted facts employed in the UMass IESL system for TAC KBP 2013 (Singh et al., 2013). The Universal Schema pattern score for a surface pattern *pat* and TAC KBP relation r is in this setting given as:

[18] We could not use the regularizer of 0.01 that is reported in Riedel et al. (2013) since this lead to numerical instability on our data set.

$$\frac{\mathbf{a_r} \cdot \mathbf{a_{pat}}}{\|\mathbf{a_r}\|\|\mathbf{a_{pat}}\|}$$

where $\mathbf{a_r}$ is the universal schema vector for the relation and $\mathbf{a_{pat}}$ that for the surface pattern. One strength of Universal Schema matrix factorization is that it can *transitively* incorporate signals from indirectly connected cells in the co-occurrence matrix.

4.5. Ranking-based Evaluation

Evaluation is done on the ranking quality according to TAC KBP gold annotations (Ji et al., 2010) of extracted facts from all TAC KBP queries from 2009-2011 and the TAC KBP 2009-2011 corpora. The queries consist of 298 query entities with types *PERSON* or *ORGANIZATION*; there are 42 relations to be considered. First, candidate sentences are retrieved in which the query entity and a second entity with the appropriate type are contained. Candidate sentences are then used to provide answer candidates if one of the patterns – extracted from the training data – matches. The answer candidates are ranked according to the score of the matching pattern. If several patterns match, the score of the highest scored pattern is assigned.

The basis for pattern extraction is the noisy DS training data used in our submissions to TAC KBP 2012 and 2013 (Roth et al., 2012, 2013) and described in detail in Chapter 5.7.1. The retrieval component described in Chapter 5.3 is used to obtain sentence and answer candidates, which are then ranked according to their respective pattern scores.

This ranking is evaluated using the TAC KBP gold annotations[19]. The basis of evaluation consists of 38,939 response candidates with matching patterns, with the corresponding facts ranked according to the score of the best pattern match. 951 of the response candidates are correct according to the gold annotation, 38 (out of 42) relations have at least one correct response candidate. Evaluation results are reported as averages over per-relation results of the standard ranking metrics mean average precision (*map*), geometric map (*gmap*), precision at rank 5 and at rank 10 (*p@5*, *p@10*). *Map* and *gmap* provide metrics over the entire ranking quality of the whole

[19]Note that those annotations are a result of pooling and therefore incomplete and under-estimating precision. However, they allow for a relative comparison of ranking quality.

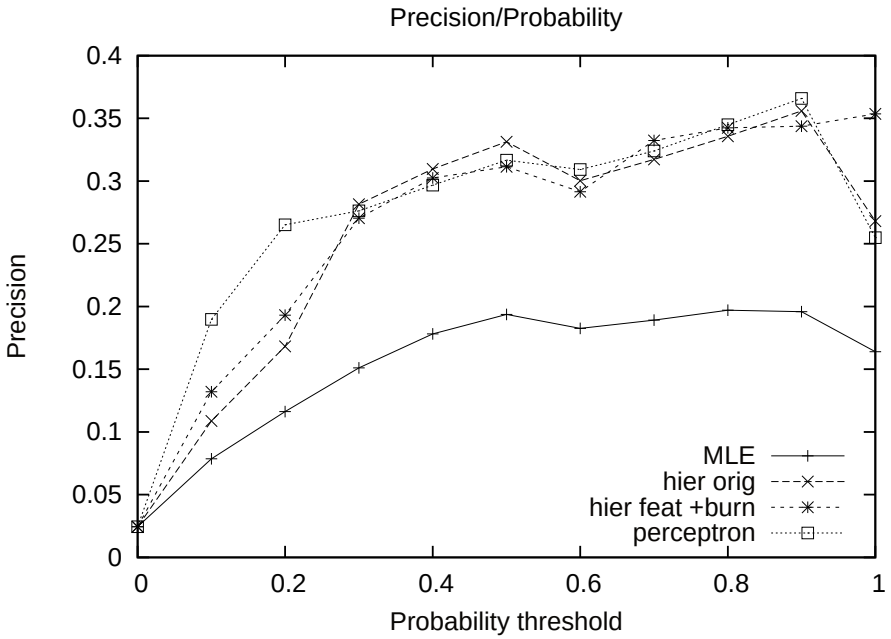


Figure 11: Precision at probability thresholds.

evaluation set (see the definition on page 51). Precision at 5 and at 10 (*p@5*, *p@10*) are included for reference – however, these metrics disregard most of the evaluation set and give a coarser picture.

The hierarchical topic model has originally been evaluated against maximum likelihood estimation by comparison of precision/probability curves (Alfonseca et al., 2012). However, note that in theory the precision values at probability thresholds can be increased (at the expense of recall) also by methods that generally lower relation probabilities without improving the overall ranking quality. While we include a precision/probability evaluation (Figure 11) for the hierarchical topic models, we focus on comparison of ranking measures (Table 5) to consider recall as well as precision.

In Table 5, the result of ranking the facts randomly (assigning uniform weight) is included as an uninformed baseline. Another simple baseline is pattern weighting by maximum-likelihood estimator (*MLE*), which scores patterns by the relative frequency of their occurrence with a certain relation. For the following methods, the model score that a certain pattern actually expresses a particular relation is weighted with the relative frequency. The original hierarchical topic model (*hier orig*) as described in

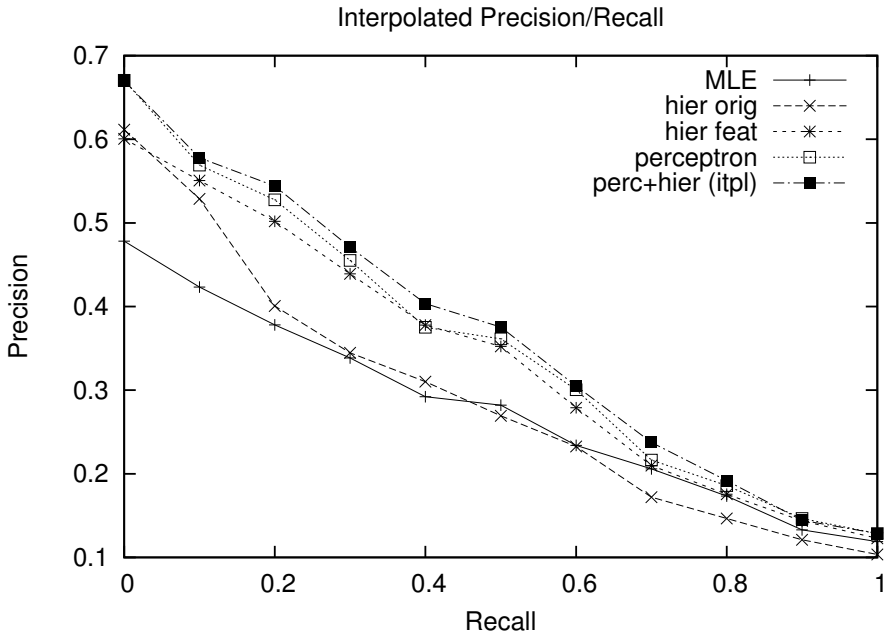


Figure 12: Precision at recall levels.

(Alfonseca et al., 2012), increases the scores under most metrics, however the increase is only significant for p@5 and p@10. Additional substantial improvements were obtained by averaging counts over the last 10 training iterations, which is known as burn-in (*hier orig +burn*). Ranking quality was significantly improved under most measures by including features (bi-grams) into the hierarchical model (*hier feat +burn*) as described in Section 4.1.2. Good results are also obtained by Universal Schema trained with BPR (*USchemaBPR*). It is worth mentioning that training Universal Schema by attempting to factorize the matrix directly (*USchemaDirect*), treating unobserved cells as negative, performs worse than the maximum likelihood estimator. This points to the importance of allowing flexibility for modeling unobserved parts of the data. The overall best results obtained by a single model are those of the perceptron learner (*perceptron*) as described in Section 4.2.

For the noise models developed in this work, combination of them leads to further improvements under the metrics. It is interesting to see that the model combinations both by non-dominated sorting *perc+hier (pareto)* as well as by uniform interpolation *perc+hier (itpl)* give an increase in ranking quality. The simpler interpolation scheme

Average Precision measures for every entity pair i relevant to a relation q ranked at $r_{i,q}$ the precision up to its rank. The per-entity-pair score

$$AP_{i,q} = \frac{\text{relevant entity pairs } j \text{ with } 1 \leq r_{j,q} \leq r_{i,q}}{r_{i,q}}$$

is combined to a per-relation score, by averaging over all scores of the set of relevant entity pairs Rel_q:

$$AP_q = \frac{1}{|Rel_q|} \sum_{i \in Rel_q} AP_{i,q}$$

Mean Average Precision combines these scores for the set Q of all relations by their arithmetic mean:

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP_q$$

Geometric Mean Average Precision uses the geometric mean instead:

$$GMAP = \sqrt[|Q|]{\prod_{q \in Q} AP_q}$$

Figure 13: Ranking-based evaluation measures.

generally works best. Figure 12 shows the Precision/Recall curves of the novel noise reduction methods developed in the previous chapter and their linear interpolation. On the P/R curve, the linear interpolation is equal to or better than the single methods on all recall levels.

4.6. End-to-End Evaluation

We evaluate the extraction quality of the induced *perc+hier (itpl)* patterns in an end-to-end setting. We use the evaluation setting of Surdeanu et al. (2012) and the results obtained with their pipeline for multi-instance multi-label relation extraction system (MIMLRE) and their re-implementation of the multi-label at-least-one perceptron (MultiR) of Hoffmann et al. (2011) as a point of reference.

In Surdeanu et al. (2012) evaluation is done using a subset of queries from the TAC KBP 2010 and 2011 evaluation. The source corpus is the TAC KBP source corpus and a 2010 Wikipedia dump. In Surdeanu et al. (2012) only those answers are considered in

method	map	gmap	p@5	p@10
uniform weights	.095	.033	.047	.058
USchemaDirect	.177	.053	.153	.124
MLE	.253	.142	.263	.232
hier orig	.270	.158	.353*	.297*
hier orig +burn	.286	.181	.379*	.300*
hier feature	.312$^{\dagger *}$	.199$^{\dagger *}$	.347*	.303*
hier feature +burn	.318$^{\dagger *}$	.205$^{\dagger *}$	.363*	.321*
USchemaBPR	.327$^{\dagger *}$	.207$^{\dagger *}$	.416*	.318*
perceptron	.330$^{\dagger *}$	.210$^{\dagger *}$	.379*	.337*
perc+hier (pareto)	.340$^{\dagger *}$	**.220**$^{\dagger *}$	.400*	.340*
perc+hier (itpl)	**.344**$^{\dagger *}$	**.220**$^{\dagger *}$	**.426**$^{\dagger *}$	**.353**$^{\dagger *}$

Table 5: Ranking quality of extracted facts. Significance (paired t-test, $p < 0.05$) w.r.t. *MLE*(*) and *hier orig*(†).

scoring that are contained in a list of possible answers from their candidates (reducing the number of gold answers from 1601 to 576 and thereby increasing the value of reported recall considerably).

For evaluating our patterns, we take the same queries for testing as Surdeanu et al. (2012). As the document collection, we use the TAC KBP source collection and a Wikipedia dump from 07/2009 that was available to us. From this document collection, we use the retrieval pipeline described in Chapter 5.3 and take those sentences that contain query entities and slot filler candidates according to NE-tags. We filter out all candidates that are not contained in the list of candidates considered in Surdeanu et al. (2012) and use the same reduced set of 576 gold answers as the key. We tune a single threshold parameter $t = .3$ on held-out development data and take all patterns with higher scores. Table 6 shows that results obtained with the induced patterns compare well with state-of-the-art relation extraction systems.

Here and in the following, the evaluation metrics are defined with respect to answer entities (*slot fillers*) that stand in one of the TAC relations with one of the query entities:

method	Recall	Precision	F1
MultiR	.200	.306	.242
MIMLRE	.314	.247	.277
perc+hier (itpl)	.248	.401	.307

Table 6: TAC Scores on Surdeanu et al. (2012) queries.

$$\text{precision} = \frac{|\{\text{relevant slot fillers}\} \cup \{\text{returned slot fillers}\}|}{|\{\text{returned slot fillers}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant slot fillers}\} \cup \{\text{returned slot fillers}\}|}{|\{\text{relevant slot fillers}\}|}$$

$$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

4.7. Illustration: Top-Ranked Patterns

Figure 14 shows top-ranked patterns for **per:title** and **org:top_members_employees**, the two relations with most answers in the gold annotations. For maximum likelihood estimation the score is 1.0 if the pattern occurs only with the relation in question – this includes all cases where the pattern is only found once in the corpus. While this could be circumvented by frequency thresholding, we leave the long tail of the data as it is and let the algorithm deal with both frequent and infrequent patterns.

One can see that while the maximum likelihood patterns contain some reasonable relational contexts, they are less prototypical and more prone to distant supervision errors. Note, that while the MLE patterns are often longer than the noise-reduced ones, this does not mean that they capture *more* variability, but rather *less*: correct long patterns are mostly infrequent lucky hits in the training data, which are very unlikely to recur with different argument pairs in the test data. The patterns scored high by the proposed combination generalize better, variation at the top is achieved by re-combining elements that carry relational meaning (*"is an"*, *"vice president"*, *"president director"*) or are closely correlated to the particular relation.

per:title, MLE

*[**ARG1**] , a singing [**ARG2**]*

[ARG1**] Best film : Capote (as [**ARG2**]*

*[**ARG1**] Nunn (born October 7 , 1957 in Little Rock , Arkansas) is an American jazz [**ARG2**]*

[ARG2**] Kevin Weekes , subbing for a rarely rested [**ARG1**]*

*[**ARG1**] Butterfill FRICS (born February 14 , 1941 , Surrey) is a British [**ARG2**]*

per:title, perc+hier (itpl)

*[**ARG1**] , is a Canadian [**ARG2**]*

*[**ARG1**] Hilligoss is an American [**ARG2**]*

*[**ARG1**] , is an American film [**ARG2**]*

*[**ARG1**] , is an American film and television [**ARG2**]*

[ARG1**] for Best [**ARG2**]*

org:top_members_employees, MLE

*[**ARG2**] remained chairman of [**ARG1**]*

[ARG2**] asks the ball whether he and [**ARG1**]*

*[**ARG2**] was chairman of the [**ARG1**]*

[ARG1**] , Joe Lieberman and [**ARG2**]*

[ARG1**] 's responsibility to pin down just how the government decided to front $ 30 billion in taxpayer dollars for the Bear Stearns deal , " Chairman [**ARG2**]*

org:top_members_employees, perc+hier (itpl)

*[**ARG2**] , Vice President of the [**ARG1**]*

*[**ARG1**] Vice president [**ARG2**]*

*[**ARG1**] president director [**ARG2**]*

*[**ARG1**] vice president director [**ARG2**]*

*[**ARG1**] Board member [**ARG2**]*

Figure 14: **Top-scored patterns for maximum likelihood (MLE) and the interpolation (perc+hier itpl) method.** Patterns that are judged to be wrong or ambiguous are marked by *.

4.8. Summary

The high ratios of noise in distant supervision training data, estimated to lie between 30% and 60% (see Chapter 3.1), make noise reduction, the filtering or suppression of false positive matches, necessary. Previously proposed generative and discriminative models to this problem had their specific shortcomings: Generative models (Alfonseca et al., 2012; Takamatsu et al., 2012) only operated on pattern level and did not model

the features associated with contextual patterns. For the long tail of infrequent patterns, the prediction whether they are good or noisy matches had to correspond to an uninformed guess. Previous discriminative models (Surdeanu et al., 2012; Hoffmann et al., 2011), while feature-based, were designed around a multi-class objective function that made explicit negative training necessary. Since knowledge bases like Freebase are incomplete and do not contain *negative* facts, negative training must be heuristically approximated, for example by sampling from the corpus, and it is not clear how much of it to include. In this chapter we have addressed both issues by proposing an extended topic model that includes features, and a perceptron learner that employs a ranking function (instead of a multi-class classifier) to give high weight to both the distant supervision signal and to the noise in the training data. We demonstrated the effectiveness of both methods and showed that further improvements can be obtained by a combination of the two. State-of-the-art extraction performance is achieved both when measuring the ranking quality of extracted facts, as well as in a query-driven end-to-end setting.

5. End-To-End System

Relation extraction is often described as the task of deciding for an argument pair in a context whether it expresses a relation or not. Models for deciding this question are trained, while the questions asked are for example: How to capture the fact that the training data is noisy (see Chapter 3.2)? How can correlations between relations be used to train better prediction models (Surdeanu et al., 2012)? Mostly, the settings in which such approaches are evaluated focus on the effect of a particular modeling strategy and try to isolate it as much as possible from other influences. Often in published work, the proposed models are compared with baselines that just leave out a particular addition, in order to show the impact of a particular employed strategy: In this line, Riedel et al. (2010) use SampleRank for all their experiments and evaluate it with and without an at-least-one; Surdeanu et al. (2012) focus on the impact of their relation modeling strategy (rather than their candidate retrieval step), by restricting the evaluation to answer candidates returned by their system and effectively rescaling recall. Such an approach is certainly inspiring and can show how some difficult problems can be solved. However, such evaluations leave the question unanswered how much an end-to-end relation extraction system would effectively benefit from the suggested improvements.

The tasks tackled in this dissertation are motivated by the TAC KBP benchmark[20], which aims at giving a realistic picture of not only precision but also of recall of relation extraction systems on big corpora, and is therefore an advancement compared with many other evaluations done for relation extraction that are often precision oriented (Suchanek et al., 2007) or restrict the evaluation key to answers from a fixed candidate set (Surdeanu et al., 2012) or to answers contained in a data base (Riedel et al., 2010). The English slot filling task of TAC KBP requires participants to extract relational information about query entities of the type *person* or *organization* from a large text corpus, and to fill in missing information about the queries in a knowledge base (see Chapter 2.2). At the center of the TAC KBP slot filling task lies the relation detection task; however, steps like document retrieval, finding and disambiguating potential query or answer matches can also have a significant impact on performance. In general,

[20] `http://www.nist.gov/tac/about/index.html`

several challenges are connected to this task:

1. Retrieving all documents and sentences from the text collection where relevant information is stored.

2. Mapping the human readable task definition to a machine readable representation.

3. Modeling both the contexts that express a relation as well as possible relation arguments.

4. Generating training data for machine learning algorithms.

5. Dealing with redundancy and ambiguity.

Since TAC KBP slot filling is formulated by stating a well-defined information need, it is designed to shed light on the question of which approaches and steps in a pipeline are most beneficial to solving a query-driven relational extraction task. Similarly to the classical TREC evaluation campaigns in document retrieval, TAC KBP aims at approaching a true recall estimate by pooling, i.e. merging the answers of a time-limited manual search with the answers of all participating systems. The pooled answers are then evaluated by human judges. We think that rather than custom evaluations, a public benchmark such as TAC KBP provides a suitable testbed for studying the influence of modeling decisions on performance.

Two dimensions constituting qualities of good research should be more emphasized in the field of relation extraction:

1. *Overall quantification of problems and effects:* As an example, an analysis of the TAC training data in Surdeanu et al. (2012) showed that only 2.8% of the training data actually exhibited the label overlap that was additionally modeled in their training approach. The resulting method is successful in mitigating the negative effect of label overlap and is certainly mathematically interesting. Still we think it is justified to ask whether this research problem is the most pressing in a field where state-of-the-art systems struggle to reach 40%$F1$-score in a realistic evaluation scenario such as TAC KBP.

In the following chapter, we aim at giving a clear picture of where potentials and problems lie that have a major impact on performance. Central to this point is also the recall analysis for identifying how much of the future potential in relation extraction should be sought in relational modeling vs. identification of context and argument candidates.

2. *Occam's razor:* More attention should be placed on the question of what would be the *simplest* approaches and representations that are successful in relation extraction. Not only do simple approaches usually shed more light into the actual nature of a problem, often they also lead to better performance. We believe that most current approaches to relation extraction can be simplified on many levels, while increasing their performance. An example is the use of dependency representations: Dependency analysis, a complex task in itself, is often (e.g. Mintz et al. (2009); Surdeanu et al. (2012)) added to a relation extraction system without comparing its impact with simpler representations, such as plain surface strings. Indeed, a comparison reveals that with current methods, a plain surface representation does seem to lead to at least equally good results (see e.g. Chapter 5.7.3 and Alfonseca et al. (2012); Illig et al. (2014)).

   As another example, Bunescu and Mooney (2007) have argued (referring to Ray and Craven (2005)) that at-least-one learning can be made obsolete by a simple scheme, namely appropriately weighting the cost function of a standard SVM classifier. A direct comparison of the two approaches, however, has not been performed in the context of relation extraction.

While it is certainly not possible to re-implement and cross-evaluate all possible decisions for a relation extraction system, a top-performing end-to-end system has to be maximally clear in designs on all levels of the pipeline. In the present Chapter of the thesis, we will outline the decisions taken for our TAC KBP system and point to interesting observations and conclusions that can be drawn from this setting.

5.1. The RelationFactory System

It is a big advantage of TAC KBP that the end-to-end setup, from the query through retrieval of candidate contexts and judging whether a relation is expressed, to normalizing answers and putting them into a knowledge base is realistic. At the same time, the task is very complex and may involve too much work overhead for researchers only interested in a particular step in relation extraction such as matching and disambiguation of entities, or judging relational contexts. To truly advance the state of the art in relation extraction, the software developed in the context of this dissertation is made open source to the research community in form of the *RelationFactory* system, a fast, modular and effective relation extraction system[21]. *RelationFactory* was the system used in the TAC KBP 2013 English Slot-Filling participation by the Spoken Language Systems at Saarland University (LSV) and was top-ranked (out of 18 systems) in the TAC KBP 2013 English Slot-filling benchmark (Surdeanu, 2013). An early version of this system was also used as the LSV 2012 slot filling system (Roth et al., 2012).

We believe that *RelationFactory* provides an easy start for researchers interested in relation extraction, and we hope that it may serve as a baseline for new advances in knowledge base population. When developing *RelationFactory*, special care was taken to achieve modularity and to adhere to design principles that facilitate changing, extending and testing the software. Those design principles conform to what is known as the *Unix philosophy*.[22] For *RelationFactory*, this philosophy amounts to a set of modules that solve a certain step in the pipeline and can be run (and tested) independently of the other modules. For most modules, input and output formats are

[21] `https://github.com/beroth/relationfactory`

[22] One popular set of tenets (Gancarz, 2003) summarizes the *Unix philosophy* as:

1. Small is beautiful.
2. Make each program do one thing well.
3. Build a prototype as soon as possible.
4. Choose portability over efficiency.
5. Store data in flat text files.
6. Use software leverage to your advantage.
7. Use shell scripts to increase leverage and portability.
8. Avoid captive user interfaces.
9. Make every program a filter.

column-based text representations that can be conveniently processed with standard Linux tools for easy diagnostics or prototyping. Data representation is compact: the system is designed in such a way that each module ideally outputs one new file. Because of modularization and simple input and output formats, *RelationFactory* allows for easy extensibility, e.g. for research that focuses solely on novel algorithms at the prediction stage.

The single modules are connected by a makefile that controls the data flow and allows for easy parallelization. *RelationFactory* is highly configurable: new relations can be added without changing any of the source code, only by changing configuration files and adding or training respective relational models.

Furthermore, *RelationFactory* is designed to be highly scalable: Thanks to feature hashing, large amounts of training data can be used in a memory-friendly way. Predicting relations in real-time is possible using shallow representations. Surface patterns, n-grams and skip-n-grams allow for highly accurate relational modeling, without incurring the cost of resource-intensive processing, such as parsing.

5.2. System Components Overview

The pipeline is a two-stage pipeline with (1) a candidate generation stage, consisting of document retrieval and sentence filtering based on named-entity type checking and query matching, and (2) a candidate validation stage, consisting of several modules that decide (typically based on the relational context) whether a candidate indeed expresses the relation or not. Figure 15 shows a simplified data-flow diagram of the prediction pipeline, Figure 16 shows the layout in more detail, including the most important modules and resources used.

The system starts with the query as provided by TAC and expands the entity name to possible other name variations of the query entity (see Chapter 5.3.1). Wikipedia link statistics and other heuristics are used for query entity expansion. The original query and selected query variants are then used to retrieve indexed documents that may contain information about the entity (Chapter 5.3). From the retrieved documents those sentences are filtered out that contain possible slot filler candidates for any of the sought relations (Chapter 5.3.4). Candidate sentences must contain a reference (name variant) to the query, and a token sequence of the appropriate slot type. Since some of

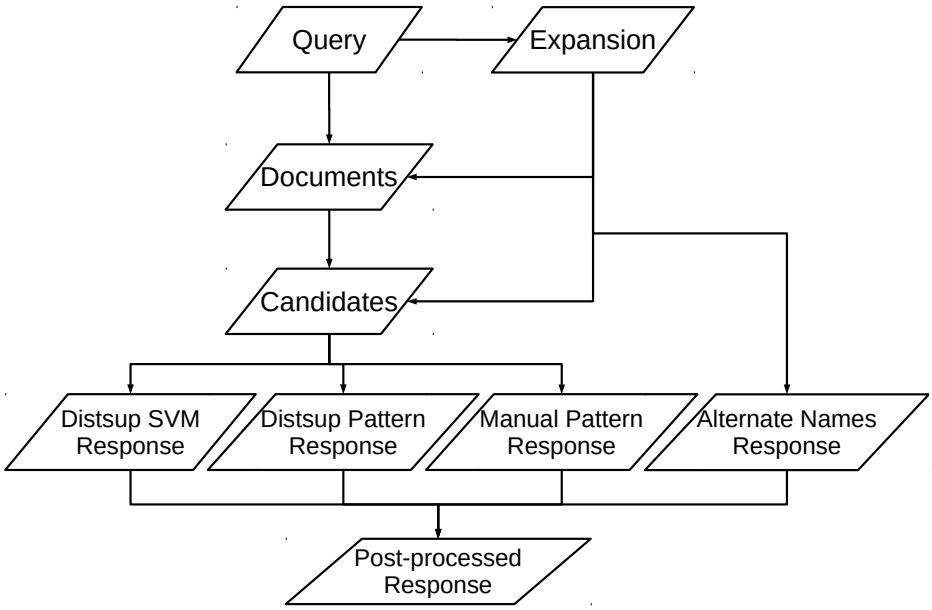


Figure 15: **Simplified data-flow of the relation extraction system.** Parallelograms depict data outputs/inputs at different stages of the pipeline. The inputs leading to the *candidates* form the *candidate-generation* stage, the inputs leading to the *post-processed response* form the *candidate-validation* stage.

the slot types are non-standard (e.g. *JOB-TITLE*, *CRIMINAL-CHARGES*), Freebase is mined for lists of appropriate entities for that type. Optionally, *JOB-TITLE*s are associated with co-occurring *ORGANIZATION*s to disambiguate whether different answer candidates for the relation `per:title` pertain to the same or to different jobs.

A series of predictors is used to judge whether *candidates*, sentences with word spans marked as potential arguments for a relation, indeed express the respective relation. Features are extracted from the candidate sentences and the instances are judged by binary per-relation SVM classifiers (Chapter 5.7). Patterns extracted from distant supervision data by scoring based on noise reduction methods are matched directly on the sentence surface token strings (Chapter 5.8). Another (high precision, low recall) module matches hand-crafted relation specific patterns (Chapter 5.5). A separate module collects all name variants of the query in the retrieved documents and returns

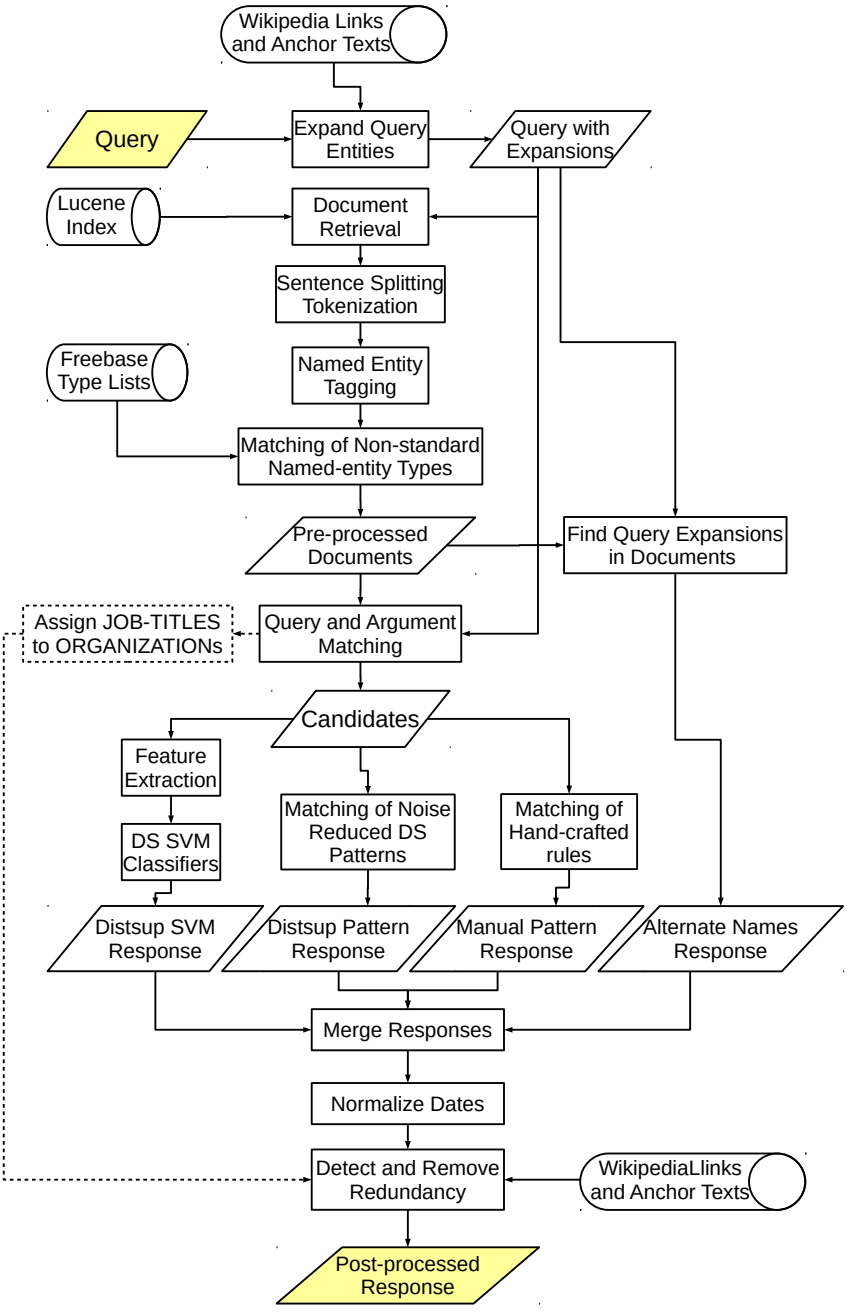


Figure 16: **Detailed schematic view of the relation extraction system (without training).** Parallelograms depict data outputs/inputs, rectangles depict modules and cylinders depict resources.

```
<query id="SF503">
  <name>Badr Organization</name>
  <docid>NYT_ENG_20070621.0157.LDC2009T13</docid>
  <enttype>ORG</enttype>
  <nodeid>NIL036</nodeid>
</query>
```

Figure 17: Query entity as provided by TAC.

them as responses for the relation `per:alternate_names`.

All responses from classifier, pattern matching and the alternate names module are then merged and post-processed to match the task-specific guidelines (Chapter 5.10): Redundant answers are removed by a mechanism similar to that used in entity expansion (based on Wikipedia link anchor text statistics), cut-offs are applied to the number of answers (e.g. for single-slot types) and dates are normalized.

The design of the single modules will be described in the subsequent sections of this chapter, their performance will be evaluated both in isolation and with respect to their impact to the performance of the end-to-end system.

5.3. Retrieval and Query Entity Matching

The extraction process starts with query definitions as shown in Figure 17 (TAC KBP query `SF503`). The query *name* is a surface form that would be used to refer to the query entity in a neutral context (e.g. in a Wikipedia infobox) and may be ambiguous.[23] The document identifier (*docid*) provides the identifier for a document where the *name* is contained and denotes the intended entity. The entity type (*enttype*) indicates whether the entity is a person or an organization and consequently which relations are appropriate. The node identifier (*nodeid*) indicates whether the entity is already contained in the initial knowledge base extracted by TAC from infoboxes of a 2009 Wikipedia dump, and would point to the corresponding entry in that knowledge base. If the entity is not yet contained in the KB, a new entry identifier is created for it, with a "`NIL`" prefix and with an appended running number.

[23] Highly ambiguous query names are avoided by the task organizers, however.

5.3.1. Expansion Schemes

For extracting the slots for all relations, it is vital to find the contexts in which possible slot fillers can be found. This in a first step means finding the appropriate documents that deal with the query entity, in a second step finding all occurrences of references to the query entity in that document, and in a third step finding candidates for possible other entities that could stand in one of the sought relations with the query entity.

Finding name variations is an important task that plays a role in the first two of the above mentioned steps: If a document does refer to the query entity only in non-standard surface forms, and the query entity is not recognized, all information in that document is lost for further steps in the pipeline. Furthermore, automatically adding name variations to the queries may, similar to automatic relevance feedback in classic IR research (Rocchio, 1971; Salton and Buckley, 1997), disambiguate the original query term and lead to a better ranking of documents. Within a retrieved document that contains information about the query, alternative surface forms (aliases) of the query are important for matching all contexts in which the query is mentioned, and which may contain an answer.

For our relation extraction system we use three mechanism to create aliases:

1. Alias generation based on Wikipedia link anchor text statistics (*Wiki-link*).

2. For persons: adding the last name only.

3. For organizations: adding variants based on possible types of business entities.

We use the expansions for retrieval and for matching directly, i.e. we do not use any other entity linking or disambiguation strategies.

Wiki-link expansion. The name of a TAC KBP query entity is expanded by a translation model based on Wikipedia anchor text, inspired by our work on cross-language information retrieval (Roth and Klakow, 2010) – however, instead of translating a query from one language to another, the query is "translated" into variants within the same language. The advantage of using anchor text rather than e.g. Wikipedia redirects is that anchor text captures a wide range of variations as they occur in actual sentences.

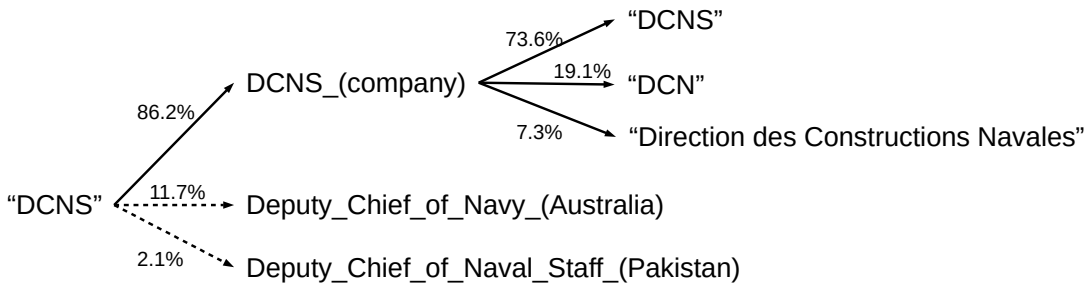


Figure 18: **Query expansion using Wikipedia link anchor text statistics.** In the first step, the most likely Wikipedia article is chosen, based on the frequency with which a surface form is linked to it. In the second step, link anchor texts for that article are returned.

We use count statistics from links in Wikipedia. For each link we call *anchor* the text that is marked as a link, and *target* the Wikipedia page it points to.[24] Given a query name (e.g. *"DCNS"*), we find the most likely Wikipedia page that is the target of links with this anchor text (in the example the Wikipedia page with the same name). From this intermediate representation, the most frequent anchors for links with this target are taken as expansions, see Figure 18 . Table 7 shows examples of expansions. We impose the following two constraints for Wiki-link expansions: (1) only links are considered that occur at least 2 times (2) only the ten most frequent expansions are used as aliases.

Computing the set of aliases A for a query q can be summarized as follows:

1. For a query q, that Wikipedia article page (wp) is selected to which q is most likely linked.

2. For this article wp, the top-n link anchor texts are returned:

$$A(q) = \{alias : P(alias|wp) \in topn(wp) \wedge wp = \arg\max_{wp} P(wp|q)\}$$

Where

$$P(alias|wp) = \frac{n(alias, wp)}{\sum_{alias} n(alias, wp)}$$

[24]Redirects are resolved when creating the statistics.

Original query	Wikipedia link anchor text expansions	Per: last name / Org: suffixes
Ali Akbar Khan	Utd. Ali Akbar Khan, Ustad Ali Akbar Khan	Khan
Adam Gadahn	Azzam the American, Adam Yahiye Gadahn	Gadahn
Augustus Hawkins	Gus Hawkins, Augustus F. Hawkins	Hawkins
Nancy Kissel	Murder of Robert Kissel, Robert Kissel	Kissel
DCNS	Direction des Constructions Navales, DCN, ...	DCNS Ltd, DCNS Corp, ...
STX Finland	Kvaerner Masa Yards, Aker Finnyards, ...	STX Finland Ltd, ...
Badr Organization	Badr Brigade, Badr Organisation, Badr Corps, ...	Badr Organization Ltd ...
Oxford University Press	Oxford Clarendon Press, Oxford, OUP, ...	Oxford University Press Ltd ...

Table 7: **Examples of query expansions.** The expansion of *Nancy Kissel* is an example of a wrong expansion to thematically related entities. The vast majority of query expansions is, however, beneficial. Note that the suffix heuristic over-generates alternate names – however, this poses no ambiguity problem since wrong suffix expansions do usually not match other entities. Underlined: Aliases removed by linkback requirement.

and $topn(wp)$ are the n highest probabilities for $P(alias|wp)$ given wp.

It can be observed that names are expanded to link texts that are frequent but not specific to the entity (e.g. "Oxford") or aliases that can be highly ambiguous (often acronyms such as e.g. "OUP" for "Oxford University Press"). In order to avoid translations to surface forms that mainly denote other entities, we include a second, more precision-oriented expansion scheme: Only those aliases are retained for which the most frequently co-occurring Wikipedia page is the same as for the original query name, i.e. for which the following condition holds:

$$\arg\max_{wp} P(wp|q) = \arg\max_{wp} P(wp|alias)$$

We call this double-sided checking *linkback*-filtering. Table 7 shows that for the example query "Oxford University Press" some wrong expansions are removed.

Person last name expansion. For entities of type *PERSON* the system also adds the last name (i.e. last token) to the aliases. However, single-token aliases for persons are not used in the document retrieval step, but only for matching after retrieval. That is, for retrieval only rather unambiguous surface forms are used. However, once it is

established that a document deals with the sought entity, also the less specific last name only is used.

Organization suffix expansion. For queries of type *ORGANIZATION*, additional expansions are generated by augmenting the original name by common suffixes that indicate types of business entities (taken from a list in Wikipedia; e.g. *Ltd*, *Corp*). Note that adding suffixes merely results in obtaining longer (more exact) matching token sequences for the entities, but does not retrieve more documents or match more occurrences, since the shorter original name is always included in all longer ones. The list of organization suffixes is in Appendix A.2.

5.3.2. Document Retrieval

Document retrieval is a vital step of the pipeline. An Apache Lucene[25] index is used for it, the aim is to obtain all, or at least sufficiently many, documents containing information about the query entity. The query entity may be expressed in one of its alias forms in the documents: However, just using all aliases leads to ambiguity and precision problems as too unspecific alias forms may be contained in the expansion. Therefore, we retrieve documents by using the original query name and one query expansion that correlates most highly with it. This expansion is selected from the aliases by high point-wise mutual information, if that value is positive. The Lucene query is built up in the following way:

1. Add the original name to the query.

2. For each alias that is not a substring or superstring of the original query name, compute the point-wise mutual information (PMI) with the original name on the document collection. Add (with OR) the alias with the highest PMI, if the PMI is positive. The score is computed according to the following formula:

$$PMI(query, alias) = log \frac{P(query, alias)}{P(query) \cdot P(alias)}$$

$$P(query, alias) = \frac{n(query, alias)}{N}$$

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

expansion	Recall	Prec	F1
none	0.1704	0.3545	0.2302
linkback	0.3115	0.3314	0.3212
all	0.3315	0.3118	0.3213

Table 8: Influence of the different expansion schemes on end-to-end performance, basic modules (SVM classifier, hand-crafted patterns and alternate names; evaluated on 2012 TAC data).

$$P(query) = \frac{n(query)}{N}$$
$$P(alias) = \frac{n(alias)}{N}$$

with $n(query)$ and $n(alias)$ the number of documents in which the query (or alias) occurs, $n(query, alias)$ being the number of documents in which query and alias co-occur together, and N the overall number of documents in the corpus.

3. If there are no documents returned by the query obtained so far, use the following back-off mechanism: Retrieve the highest ranked document for a (logical *OR*-) query containing all aliases.

The document threshold is set to a maximum of 500 retrieved documents per query.

5.3.3. Evaluation of Query Expansion

We evaluate the query expansion mechanism on three levels, which we will discuss in the reverse order of their occurrence in the pipeline:

1. End-to-end: What is the overall impact on performance of the relation extraction system?

2. Candidate level: How do the expansions help to find sentences containing correct answers?

3. Document retrieval: What is the impact on finding good documents and ranking them high?

expansion	Recall	Prec	F1
none	0.1478	0.3645	0.2103
linkback	0.2589	0.3324	0.2911
all	0.2669	0.3127	0.2880

Table 9: Influence of the different expansion schemes on end-to-end performance, classifier module only (evaluated on 2012 TAC data).

expansion	Recall
none	0.3442
linkback	0.5885
all	0.6131

Table 10: Recall on candidate level for the different expansion schemes (TAC KBP 2012 data).

The end-to-end performance of the different expansion schemes is shown in Tables 8 and 9. We evaluate the query expansion in conjuction with the basic prediction components without noise reduction (SVM classifier, hand-crafted patterns and alternate names matcher, see Chapters 5.7, 5.5 and 5.9) and with the SVM classifier alone (as the strongest single component).[26] In both settings the positive influence of the expansion is striking: Recall is almost doubled (relative increase of 80%), while Precision dropping only by $4-5\%$ absolute points ($11-14\%$ relatively). While it is evident that the linkback filtering retains much of the precision, this comes at the expense of recall compared with the full expansion – which of the two schemes to use finally depends on the optimal point on the precision/recall curve in combination with the subsequent relation extraction modules.

For measuring performance on candidate level, we count how many correct answers are contained in the sentences passed to the relation validation modules predicting the relations (e.g. classifiers or pattern matcher). This provides the upper bound of overall obtainable recall, since answers not contained in this set are irretrievably lost. Precision

[26]The scorer is set to 'anydoc' mode for all evaluations, i.e. the answers are evaluated independently of whether the document is in the gold document set or not.

would be little meaningful here, since a big number of candidates are expected to be irrelevant in any case – precision can be regarded as mainly the task of the subsequent modules, there is no upper bound to it. Table 10 shows the importance of expansion for generating candidates. Still, with the best recall being 61%, there seems to be room for improvement. Missing additional recall on the candidate level can have one of the following causes: Either a relevant document is not retrieved, the query has not been matched, the slot filler is not in the same sentence as the query match, or the slot filler candidate was not matched due to a tagging error. Only the first and second of these cases is related to query expansion, the others will be investigated in Section 5.4.

expansion	Prec	Recall	map	gmap	P@10	R@10
none	0.0932	0.8771	0.4754	0.2474	0.3725	0.4169
linkback	0.0866	0.9019	0.5290	0.3670	0.4150	0.4937
all	0.0897	0.9119	0.5354	0.3776	0.4250	0.4916

Table 11: Influence of the different expansion schemes on document retrieval (TAC KBP 2012 data).

Looking at the document retrieval evaluation (Table 11), it is interesting to note that the overall recall of documents is already quite high without the query expansions, and with them is only increased by 3.5%. However, the overall ranking quality (as indicated by *map* and *gmap*) and the quality of the top-ranked documents (*P@10, R@10*) is substantially increased. Moreover, as the big improvements for *gmap*[27] indicate, expansion is especially beneficial for hard queries that would only have very few documents ranked up otherwise: While for queries with many documents retrieved, additional documents are likely to contain redundant information, retrieving documents for hard queries is more likely to add useful candidates.

Comparing the document retrieval results with the candidate recall and end-to-end scores, the numbers suggest that while entity expansion has a solid positive impact on document retrieval, its effect on finding good candidate sentences within the retrieved documents is of even greater importance. This may be because one canonical mention

[27] using the geometric mean, *gmap* is more sensitive to improvements on queries with below average scores

in the document is enough for it being retrieved – however, each missed reference to the query entity in that document means a lost candidate, and hence a potentially lost answer.

5.3.4. Candidate Generation

From the retrieved documents, those sentences are retained that contain a mention of the query name or an alias, and a token sequence tagged with the expected slot type. We use a perceptron-trained sequence labeler (Collins, 2002) on the BBN training data (Weischedel and Brunstein, 2005) after mapping the BBN label set to the coarse-grained set as listed below:

BBN-mapped labels		
CARDINAL	CITY	COUNTRY
DATE	MONEY	ORDINAL
ORGANIZATION	PERCENT	STATE-OR-PROVINCE
POLITICAL	QUANTITY	PERSON
Extra labels		
CHARGES	CAUSE-DEATH	JOB-TITLE
RELIGION	URL	

We use the same word cluster features and implementation as described in Chrupała and Klakow (2010). The overall performance of the NE labeler on section 22 of the BBN corpus is shown in Table 12.

Precision	Recall	F-measure
91.18	92.15	91.66

Table 12: NER results on BBN section 22.

Additionally, we provide lists of typical strings for types that cannot be mapped to the BBN labels or where there is insufficient training data. At tagging time, all token sequences that match one of the list entries are tagged with the respective type. We obtain these lists by enumerating all entries of the corresponding types in Freebase. URLs are separately matched by a regular expression.

5.4. Candidate Recall Analysis

The recall values at the candidate stage are of crucial importance, since lost recall cannot be recovered by the subsequent validation modules. The influencing factors are document retrieval, query and argument matching. A recall analysis shows that while there is a good recall on the document level, a large potential lies in candidate sentence extraction.

Query expansion	document recall	candidate recall	end-to-end F1
no	0.8771	0.3442	0.2302
yes	0.9019	0.5885	0.3212

Table 13: Bottleneck candidate generation 2012 queries.

Query expansion	doc. recall	cand. recall	end-to-end F1 (exact)	end-to-end F1 (anydoc)
none	0.9216	0.3429	0.2454	0.3059
wiki	0.9408	0.4170	0.3097	0.3587
suffix	0.9216	0.3484	0.2635	0.3206
lastname	0.9216	0.4903	0.3344	0.3758
full	0.9443	0.5150	0.3714	0.4010

Table 14: Bottleneck candidate generation 2013 queries.

Tables 13 and 14 give an overview of the recall effects of query expansion and its impact on end-to-end performance. It can be seen that increased recall on candidate level has a direct positive impact on the final F1-score for both the 2012 and 2013 queries. For the 2013 queries, we give a more detailed analysis in Table 14, which shows the effect of the different expansion schemes. The biggest effect on recall comes from the simplest expansion by additionally including the last name only for persons.[28] Wikipedia anchor text expansion is overall the second most effective ex-

[28]For persons, last names and other expansions that consist of exactly one token are only included in the pipeline steps after document retrieval, since last names are potentially too ambiguous on a global scale. However, once a document dealing with a particular person is retrieved, a more relaxed matching scheme can be applied to find referring mentions.

pansion scheme, and the scheme that has the strongest influence on document recall. Adding organizational suffixes has a small overall positive influence. Interestingly, it improves end-to-end F1-score, but not candidate recall, which may point to the fact that the main contribution of this suffix expansion lies in providing answers for the `org:alternate_names` relation (Chapter 5.9) and not in improved matching of relational contexts.

error category	missing recall
Doc not retrieved	5.59%
Query not matched	10.37%
Slot not in query sentence	16.63%
Slot tag inexact	5.36%
Slot not tagged	24.85%
Other (validation)	37.17%

Table 15: **Recall analysis on 2013 data.** Error categories are ordered according to their occurrence in the pipeline.

Table 15 shows how much of the missing end-to-end recall for the 2013 main run is due to different possible causes. This fine-grained analysis is possible since the gold annotations for 2013 contain not only the correct answers, but also the exact character offsets of the judged slot fillers in the corpus files. For evaluating the amount of recall missed by our system, we take into account all annotations of contexts judged as *correct* in the annotated key, and consider the subset that was not returned as a correct answer by the system. Since the same correct answer can be expressed several times (i.e. redundantly), it may be that for one missing system answer, several contexts are taken into account for the recall analysis. Each gold context for a missed answer is then categorized by a cascade of checks of the reasons why the system missed this answer. If a context would be missed by the system for several causes (e.g. the corresponding document is not retrieved and the slot filler would not be tagged correctly), the context is assigned the error category of the first failing check. The checks are, in the applied order:

1. **Is document retrieved?**

2. **Is query matched?** The outcome determines whether a sentence is considered for further processing.

3. **Is answer in query sentence?** This tests whether the answer is in one of the sentences with the query. Our system can find answers only when this is the case, as there is no co-reference module included.

4. **Do answer tags overlap with gold answer?**

5. **Do they overlap exactly?**

6. **Other (validation).** If all previous checks are passed, the candidate has been correctly generated by the candidate generation stage, but the validation modules have failed to predict the relation.

One can see that the majority ($\sim$ 63%) of recall is lost in the candidate generation stage and 37% in the validation stage. The development of better relational classifiers (together with parameter tuning) would improve the recall during validation. The main recall loss during candidate generation is due to tagging errors (30%). In 25% of the cases the tagger missed the slot filler altogether, while in additional 5% of the cases the span found by the tagger had only inexact overlap with the slot filler from the gold answer. Since tagging is the most important step in the candidate generation pipeline, we provide further analysis in section 5.4.1. In $\sim$ 17% of the cases, a slot is not found since it is not in the same sentence as the query, and cross-sentence co-reference or reasoning would be required. Missing query matching is a cause for $\sim$ 10% of the missing recall. This indicates that the query expansion already works reasonably well. Further improvements for query expansion could be obtained e.g. by re-matching queries with new aliases predicted as slot fillers for `per:alternate_names`. With $\sim$ 90% recall on document level (see Tables 13 and 14) and being responsible for only 5% of missing slot filler recall, document retrieval performs already very well, and we do neither see the necessity nor an obvious way to improve document recall further.

5.4.1. Tagging Analysis

The above analysis revealed that tagging is the most important factor for candidate recall in the TAC KBP domain. In the following, we provide additional recall analysis for slot filler tagging. In Chapter 8, we will show how inducing and using argument types can help in the food domain, where the tagging problem is even more severe since all entities belong to a type usually not covered by taggers.

relation / expected slot tag	missed tagging recall
per:age / CARDINAL	15.5%
per:title / JOB_TITLE	12.1%
per:employee_or_member_of / ORGANIZATION	11.0%
per:children / PERSON	4.7%
per:origin / GPE:COUNTRY or NORP:NATIONALITY	4.6%
per:countries_of_residence / GPE:COUNTRY	4.6%
per:alternate_names / PERSON	4.6%
org:country_of_headquarters / GPE:COUNTRY	3.9%
org:members / ORGANIZATION	3.2%
org:stateorprovince_of_headquarters / GPE:STATE_PROVINCE	3.1%

Table 16: Percentage of missed recall attributed to different relations.

Table 16 breaks down the missed tagging recall per-relation, showing the 10 relations with the biggest percentages of missed slot fillers. One can see that there is no single relation or tag responsible for the majority of errors. Interestingly, the relation `per:age` with the required slot type `CARDINAL` (which should be quite easy to detect by looking for sequences of digits) constitutes the relation with most tagging errors. Manual inspection shows that this is due to a confusion with the `DATE` tag which seems to be given priority over the `CARDINAL` tag in many cases by the tagger.[29] `per:title` is the relation with second most tagging errors – this is not surprising since for the `JOB_TITLE` tag only list-lookup is performed and no context is taken into consideration. It is also interesting to note that while `per:title` accounts for 12.1% of overall missed

[29] An example sentence where this happens is: *Koirala died on Saturday afternoon at the age of 86.*

tagging recall, it accounts for 30.9% for the recall missed due to inexact tagging spans. For example, while *artist* and *Minister* are in the titles list, *jazz artist* and *Planning Minister* are not – here an expansion heuristic based on part-of-speech tags would be a conceivable remedy. As with the two relations discussed above, we expect different micro-effects systematically at work across the range of relations. Tags may also be in conflict with each other, that is, a tagging desirable for a certain relation may be different from the optimum the tagger was trained for on some general purpose data. Therefore, future improvements on tagging may lie in relation-specific taggers, tagger adaptation or re-tagging in order to undo decisions which are not optimal for a specific relation in question.

In this section, we have shown that both candidate generation and validation have a significant impact on end-to-end recall. A high recall in candidate generation is vital, since recall lost in this stage cannot be undone, and it therefore determines the upper bound of the achievable recall. In candidate generation, query expansion and argument type tagging are of crucial importance. The performance of the validation modules will be discussed in the next sections.

5.5. Hand-Crafted Patterns

In TAC KBP, the task is defined by a human readable task description, mostly independent of restrictions on the kind of methods to be used. The task guidelines and slot definitions contain roughly half a page of description per relation. These descriptions consist of definitions and examples that are supposed to give a human readable guidance for judging whether a relation is considered to hold in a particular context. Whatever learning algorithm is used, there has always to be a mapping or transformation of the guidelines performed by a human to some machine readable resource or algorithm.[30] The manual human effort can be e.g. a mapping to Wikipedia info-boxes or to Freebase relations, the creation of gazetteers, the annotation of training data, specific algorithmic routines, or formulation of question templates or patterns. The most straight forward approaches to capturing that human translation step are either providing hand-crafted seed patterns or manually establishing mappings to knowledge-bases such as Wikipedia infoboxes or Freebase. We found it generally to be less effort to write down a few token sequences than to identify the corresponding relational correspondence in Freebase, especially since certain sequences follow directly from the examples and definitions of the task description.

In order to keep the effort of writing seed patterns minimal, in our system we restricted the patterns to plain sequences of tokens with a general placeholder (* denoting 1 to 4 tokens) and did not use syntactic patterns that would require linguistic expertise. The patterns follow directly from the definitions and examples given in the guidelines. For example, if the guidelines contain an example sentence for *per:stateorprovince_of_birth*

Harper, born in April of 1959 in Toronto, Ontario

then a pattern to consider would be

*ARG1 , born * in *, ARG2*

[30] A system that would read task guidelines and program itself accordingly probably would be AI complete.

where *ARG1* stands for the query entity match and *ARG2* for the slot filler, the asterisk (*) is used to indicate 1 to 4 tokens. If such a pattern matches, the slot candidate is scored positive by the system. Together with the type filter on slot candidates from the previous step these patterns are of high precision, but it is obvious that they are of very limited coverage. The main use of these patterns, therefore, is to extract distant supervision training data (see Chapter 5.7), which can be seen as a form of pattern expansion.

5.6. Influence of Hand-Crafted Patterns

System / Pattern Component	Precision		Recall		F1-Score	
NYU / local patterns	47.4		9.3		15.6	
NYU / bootstrapped linear	59.2		4.6		8.5	
NYU / bootstrapped dependency	54.8		3.7		6.9	
LSV / token sequence	43.1	(49.0)	8.0	(8.4)	13.5	(14.3)

Table 17: **Comparison of the NYU hand-crafted pattern modules and the seed pattern component used in our system (LSV), on the 2012 task.** For the LSV system we give the exact evaluation of the 2012 system, and in brackets the *anydoc* and *lowercase* evaluation of the currently used system.

For quantifying the influence of the seed patterns in our system, we compare the performance of our seed patterns to the reported scores of hand-written patterns in the NYU 2012 system (Min et al., 2012a). In the NYU 2012 system there are three modules with dedicated hand-crafted patterns: A so-called *local patterns* module, that includes short patterns similar to ours, and two *bootstrapped patterns* modules, that take additional dedicated manual seed patterns as an input and iteratively add new patterns, based on corpus co-occurrences. The NYU pattern bootstrapping modules use hand-crafted seed patterns both based on token sequence and syntactic paths.

Table 17 shows the performance of the manual NYU pattern modules and that of our seed pattern module for the TAC 2012 task. It should be noted that performance of a particular module is also affected by other factors such as retrieval, argument

matching and post-processing. The performance of the seed patterns in our system corresponds roughly to that of the NYU local pattern component. As can be expected, the patterns show good precision, but lack recall.

5.7. Distant Supervision SVM Classifiers

The most contributing candidate validation component, both in terms of stand-alone F1-score, as well as F1 contribution in the ablation analysis (see the detailed analysis in Section 5.12), is the set of distantly supervised relation specific SVM classifiers. In this chapter, we describe the setup of training and applying the classifiers, and give an analysis of factors that impact the performance of the SVM classification module.

We will look at the following factors in isolation: parameter tuning, the source of the distant supervision training data (Freebase vs. seed patterns) and feature sets (shallow vs. syntactic). We will also look at the impact of a training scheme that aggregates all sentences for an entity pair, which was mainly chosen for having fewer training instances and hence shorter training time. [31]

5.7.1. Training Data

In this section, we will compare two ways of obtaining distant supervision entity pairs and training data, using Freebase vs. pairs from patterns. Training is done in a distant supervision setting, pairs of arguments that are known to stand in a particular relation are matched against a text corpus. Those sentences in which both arguments appear together are taken as positive examples, while the positive examples from the other relations are taken as negatives. We use two ways of obtaining pairs associated with a particular relation:

1. Pairs of entities that are connected with Freebase relations that correspond to TAC KBP relations.

[31] Since in this analysis some of the compared alternative settings were not included in the pool of answers for the official TAC evaluation, such settings have considerably lower scores when compared with the standard evaluation that requires exact provenance. We therefore additionally include the more robust *anydoc* evaluation for all such runs, which should also be used for all comparisons.

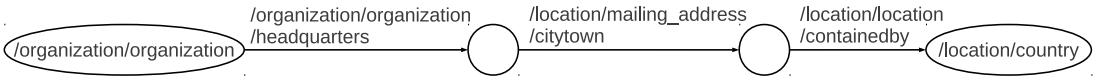


Figure 19: **Path of Freebase relation that corresponds to the TAC KBP relation** `org:country_of_headquarters`**.** Nodes correspond to Freebase entities that can be matched, they can contain restrictions on their type; edges correspond to single relations in Freebase (e.g. the relation `/location/mailing_address/citytown`). The start node is of type `/organization/organization` and corresponds to the query argument in the TAC KBP relation. The end node is of type `/location/country` and corresponds to the slot argument.

2. Pairs of entities that occur in the corpus at least once in a sentence between which a manual pattern of the respective relation matches (see Chapter 5.5).

In the first case, most TAC KBP relations correspond to joins on the Freebase database.[32] We formulate database queries on Freebase that can contain both restrictions on (binary) Freebase relations as well as on entity types. The database queries can be seen as graph configurations. See Figure 19 for an example of a graph configuration in Freebase that is mapped to a TAC KBP relation. Note, that while in the second case (pattern matches) the pattern has to match at least one context of an argument pair, also the other occurrences of that pair are considered as training input for the distant supervision classifier. Our system is the first KBP system to use patterns for generating distant supervision data in such a one-step bootstrapping setup (Surdeanu, 2013).

With these two strategies we obtain two sets of seed pairs (1) from mapping Freebase relations to TAC relations and (2) by matching seed patterns. For both methods, a threshold of up to 10.000 pairs is used per relation, the pairs are then matched against the TAC 2009 text corpora, and a maximum of 500 sentences per pair are used as training data. In order to reduce ambiguity, we consider entities only in their full canonical form (e.g. person names with first and last name) as distant supervision matches. With Freebase, 209381 argument pairs are obtained (on average 5106 per

[32]The relation *per:age* is not encoded in Freebase, as it is relative to document creation time.

j-parameters	Recall	Prec	F1	Recall *any*	Prec *any*	F1 *any*
0.1	0.0197	0.5686	0.0381	0.0246	0.7058	0.0476
1	0.1573	0.2942	0.2050	0.2440	0.4535	0.3172
10	0.2002	0.1513	0.1724	0.3564	0.2677	0.3057
tuned	0.1839	0.2749	0.2204	0.2954	0.4389	0.3531

Table 18: **SVM classifiers trained with distant supervision data from Freebase pairs.** The *j*-parameter is the cost-factor by which training errors on positive examples outweight errors on negative examples. It can be set to a uniform value for all relational predictors, or tuned for an optimal configuration of per-relation parameters. *any* stands for the more robust *anydoc* evaluation setting, where answers are considered correct or incorrect independently of their provenance in the document collection.

j-parameters	Recall	Prec	F1	Recall *any*	Prec *any*	F1 *any*
0.1	0.1178	0.3589	0.1774	0.1569	0.4751	0.2359
1	0.1553	0.2711	0.1974	0.2309	0.4007	0.2930
10	0.1682	0.1971	0.1815	0.2590	0.3016	0.2787
tuned	0.1559	0.2813	0.2007	0.2405	0.4312	0.3088

Table 19: Using data from pattern matching pairs.

j-parameters	Recall	Prec	F1	Recall *any*	Prec *any*	F1 *any*
0.1	0.0708	0.4110	0.1208	0.0884	0.5098	0.1507
1	0.2132	0.3216	0.2564	0.2871	0.4306	0.3445
10	0.2343	0.1816	0.2046	0.3783	0.2914	0.3292
tuned	0.2350	0.345	0.2795	0.2988	0.436	0.3546

Table 20: Using merged Freebase and pattern data.

relation), and 999201 training sentences are retrieved (on average 4.8 per pair). With the pairs from pattern matching, 149079 argument pairs are used (3636 per relation), with a total of 960535 training sentences (6.4 per pair). The sizes of both types of training data are hence similar.

Table 18 shows the performance of the Freebase data alone, and Table 19 that of the pattern data, and Table 20 the performance of the merger of both data sets. Since the tuned model using the merged training data was submitted to the official TAC KBP 2013 evaluation and therefore has been included in pooling and in the official key, care should be taken when comparing the results to other settings that deviate in the resulting models: there is always some negative bias for models not included in the key. Therefore, for comparison the *any*doc scores should be used, where answers are taken into account irrespective of their provenance: here the bias is in general not so strong, since all occurrences of an answer count.

After tuning, the Freebase data leads to substantially better performance than the pattern data. The precision values are similar for the tuned Freebase and patterns models, while the recall is substantially higher for the Freebase model. We therefore attribute the better overall performance of the Freebase data to a larger variety in linguistic forms captured which are not steered towards features stemming from a limited set of patterns. Adding pattern pairs to the Freebase pairs does not substantially increase the overall performance, although the amount of data is roughly doubled: When evaluated in *anydoc* mode, the performance of the Freebase data alone comes very close to the performance of the merger of Freebase and pattern data. While the distant-supervision model only using the pattern data performs not as effectively as the Freebase model, it should be noted that it is a massive improvement compared with using the manual patterns directly (with an *anydoc* $F1$ of 14.3%, see Table 17 on page 78), and can be an effective training generation strategy in situations where no knowledge base is available to seed the distant supervision training process.

5.7.2. Parameter Tuning

In this section we will outline the training and parameter tuning process. We will defer the discussion of the feature set to Chapters 5.7.3. Unless indicated otherwise, all experiments are carried out using all of the training data (i.e. from both Freebase and patterns) and the shallow skip-n-gram featureset (see Chapter 5.7.3).

We train one binary support vector machine for each of the relations using the distant supervision matches for that relation as positive data, and the matching contexts for all other relations as negative data. If the same feature vector happens to occur more

than once in the training data, and is labeled both as positive and as negative, those instances of the feature vector which are labeled as negative are removed from the training data. We group all sentences per entity pair, extract the features, sum the feature counts of all these sentences and normalize the feature vector for that pair so that the highest feature has a weight of 1.0. We use SVM^{light} [33] as the classification toolkit.

Tuning the cost-factor by which training errors on positive examples outweigh errors on negative examples (also called j-parameter in SVM^{light}) is a hyperparameter that can be crucial to performance. Moreover, experimental results suggest that simple misclassification cost tuning is superior to multi-instance learning in many settings (Ray and Craven, 2005) including relation extraction (Bunescu and Mooney, 2007).

We therefore trained three SVM configurations for each relation by setting the j-parameter to 0.1, 1.0 and 10.0, respectively. We found that the best local parameter choice (i.e. the parameter settings that produce best per-relation F1-scores) does not necessarily correspond to an optimal global (micro-average) F1-score: For example, for relations with a low precision over the whole recall range (e.g. due to errors in a previous tagging step), increasing the individual F1-score by increasing recall may have a negative overall effect. Likewise, for relations with an above average precision, it may be beneficial for overall performance to score more instances as positive than tuning for individual F1-score may result in.

To avoid these problems that arise by individually maximizing per-relation F1-scores, we use a greedy procedure to tune the per-relation j-parameters in order to optimize *global* F1-score instead. Algorithm 2 shows the pseudo-code of the global parameter-tuning. We use $\mathcal{R}$ to denote the set of relations, $j(r)$ a choice of parameter for a particular relation $r \in \mathcal{R}$, *evaluate()* a function returning the global F1-score for the current choices of $j(\cdot)$, and *evaluate(j(r)\j)* the global F1-score with a particular $j(r)$ replaced by j. The parameters are tuned with respect to performance on earlier TAC KBP slot filling queries (years 2009–2012).

As development data for parameter tuning we use the official TAC queries and keys from 2009-2012, including the additional training data provided by the organizers for 2010. The development queries contain 404 entities. We use the 2012 *document collec-*

[33] `http://svmlight.joachims.org/`, (Joachims, 1999)

Algorithm 2 Global parameter tuning. The second loop over the relations can be executed iteratively (in our setting it was executed twice).

```
for r ∈ R do
    j(r) ← 0.1
f_1 ← evaluate()
for r ∈ R do
    for j ∈ {0.1, 1.0, 10.0} do
        f̂_1 ← evaluate(j(r)\j)
        if f̂_1 > f_1 then
            f_1 ← f̂_1
            j(r) ← j
```

tion only, as it is the biggest of the collections for the included years and has some overlap with each of the collections for the other years. Using the 2012 document collection for all the queries (from different years) greatly simplifies processing (compared with using each subset of development queries with the respective document collection), but potentially produces a mismatch that may lead to generally lower development scores. When testing, we look at the performance of the 2013 queries and key with the proper 2013 document collection. Table 21 shows results for different values of the j-parameters (uniformly set to all relations), as well as the results for choosing the j-parameter values for each relation using the optimization algorithm. The first observation is that setting the parameter uniformly to $j = 1$ already gives decent results, and is only moderately increased by the optimization algorithm. This indicates on one side, that the overall setup is robust and works well when using standard settings for the classifier; on the other hand small improvements on the development data also indicate only small over-fitting.

Table 20 on page 81 shows the impact of tuning, evaluated on the 2013 queries.[34] The impact of tuning is comparable to the effect on the development data: small and consistent. Since the untuned models have not been submitted to TAC KBP 2013, and therefore have not been included in pooling the key, they should be compared with the

[34]The slightly lower score for the tuned merged model, as compared to the official submission results, stems from a small change in the retrieval step of the system after refactoring the code.

j-parameters	Recall	Prec	F1
0.1	0.0663	0.2690	0.1064
1	0.2216	0.2006	0.2106
10	0.3119	0.1086	0.1611
tuned	0.2511	0.2157	0.2320

Table 21: Scores on development data (anydoc, 2012 index for queries from 2009 - 2012).

tuned standard setting in the *anydoc* evaluation mode.

Comparing the impact of tuning on the Freebase vs. the pattern training data (Tables 18 and 19 on page 81), one can see that tuning the cost parameters has a much stronger effect on the Freebase data than it has on the pattern data. There also is a wider range of the average precision/recall ratios for the Freebase data. The underlying reason may be a higher degree of variation for Freebase in true positive rate per relation: While for patterns at least one occurrence per pair is assumed to be true (from the originally matching pattern), the Freebase pairs may or may not contain a true positive sentence. For difficult relations, the Freebase pairs may even contain a true positive only as an exception. Such difficult relations would be effectively suppressed by tuning the classifier parameters.

5.7.3. Feature Set

The feature set in the most successful run submitted to TAC 2013 is rather minimalistic. We do not include most of the features used in the TAC 2012 predecessor system (Roth et al., 2012) (e.g. argument features, distance features, Brown cluster features), but rather model context only with token n-gram-based features. When using token n-grams, we found it essential to mark whether the query (referred to as *ARG1*) or the slot filler (*ARG2*) comes first. Additionally, including *sparse* n-grams, where tokens in the middle of the n-gram were wildcarded, increased performance. For the context between *ARG1* and *ARG2*, we use n-grams up to length 3 and skip-n-grams of length 3 and 4. We model the left and right contexts outside the arguments with n-grams up to length 3 (including the corresponding wildcarded argument). Figure 5.7.3 shows

Relation: `per:origin(Adam Gadahn, U.S.)`

Candidate sentence: *One Pakistani intelligence official said he is Adam Gadahn, a California native and the first U.S. citizen to be charged with treason in 52 years.*

Feature examples:
BETWEEN_n-gram#ARG1>#,>
OUTSIDE_n-gram#ARG2>#citizen>#to>
SKIP_n-gram#native>###first>

Figure 20: **Examples of extracted features.** Each feature is first marked with the feature group it belongs to (n-gram between or outside the arguments, skip-n-gram), followed by the token sequence of the n-gram, using # as a separator. Each token is marked to indicate whether the slot filler comes left (<) or right (>) of the query.

examples of extracted features for a candidate sentence.

The idea that a parse analysis is an appropriate representation for modeling relational content has been popular (Mooney and Bunescu, 2005; Mintz et al., 2009; Sun et al., 2011; Min et al., 2012a; Plank and Moschitti, 2013), but, interestingly enough, is usually not compared with much simpler and computationally cheaper methods based directly on surface forms. The underlying (and not verified) assumption often is that relational arguments in the majority of cases correspond to syntactic arguments. In this section we compare the n-gram features introduced in Chapter 5.7.3 with the popular feature set of Mintz et al. (2009) that is identical or similar to most syntactic feature sets used in the literature.

The Mintz et al. feature set comprises two types of features, lexical and syntactic ones (see Figure 21 for illustration). The lexical features are defined as follows:

"Our lexical features describe specific words between and surrounding the two entities in the sentence in which they appear:

- *The sequence of words between the two entities*
- *The part-of-speech tags of these words*

- *A flag indicating which entity came first in the sentence*
- *A window of k words to the left of Entity* 1 *and their part-of-speech tags*
- *A window of k words to the right of Entity* 2 *and their part-of-speech tags*

Each lexical feature consists of the conjunction of all these components. We generate a conjunctive feature for each $k \in \{0, 1, 2\}$*."* (Mintz et al., 2009)

The definition of the syntactic features is:

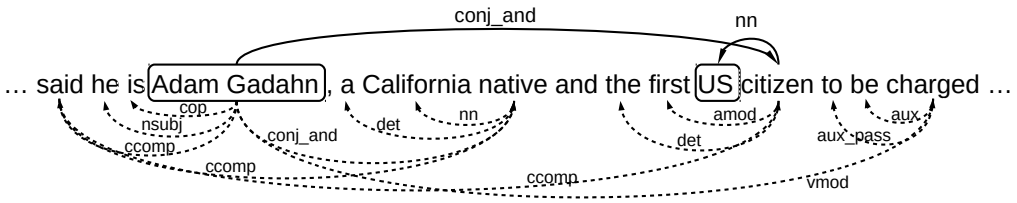
"Our syntactic features are similar to those used in Snow et al. (2005). They consist of the conjunction of:

- *A dependency path between the two entities*
- *For each entity, one 'window' node that is not part of the dependency path*

A window node is a node connected to one of the two entities and not part of the dependency path. We generate one conjunctive feature for each pair of left and right window nodes, as well as features which omit one or both of them." (Mintz et al., 2009)

The features used by Mintz are what we call *sparse features*: the features per instance are based on very specific information, which is not broken up into smaller pieces. In order for a feature to be useful in prediction, the test instance has to have exactly the same surface form (for the lexical features) or syntactic analysis. The contrast would be a *smooth* feature representation like our n-grams and skip n-grams, where overlap between instances is expressed more gradually. The assumptions in the Mintz setting are that (1) syntactic analysis provides the right kind of generalization (this assumption is implicit in Mintz et al. (2009)) and that (2) not much generalization is needed anyway as vast amounts of DS training data make more smooth feature sets unnecessary. Mintz et al. argue:

"This yields low-recall but high-precision features. With a small amount of data, this approach would be problematic, since most features would only be seen once, rendering them useless to the classifier. Since we use large amounts of data, even complex features appear multiple times, allowing our high-precision features to work as intended." (Mintz et al., 2009)



[he is] PER [,/, a/DT California/NNP native/NN and/CC the/DT first/JJ] COUNTRY [citizen to] [is $\Uparrow_{cop}$] PER [$\Downarrow_{conj\_and}$ citizen $\Downarrow_{nn}$] COUNTRY []

Figure 21: **Dependency parse (snippet) of the example sentence as used for the Mintz feature extraction, created by the Stanford parser.** Heads point to their dependents. On the top and in solid are the edges on the shortest path between the arguments, which builds the basis for the syntactic features. Below the lexical feature for $k = 2$ and the syntactic feature for *"is"* as left window node.

We will test this hypothesis by comparing both Mintz and shallow features on the TAC KBP data using the distant supervision data described in Chapter 5.7.1. On average, our data has a ratio of 8742 argument pairs (=training instances) per relation. The data of the original Mintz experiment had roughly double the size (17647 argument pairs per relation) as the most frequent Freebase relations are selected there. Nevertheless, we think that our setting still qualifies as using *large amounts of data* and may serve as a fair testbed for comparison.

Another point to consider is speed: large amounts of data require efficient ways of processing. Any costly method, like parsing, should be motivated by strong indicators of their effectiveness for a task. On our data, parsing the training data alone took 130 hours (using the Stanford parser (Cer et al., 2010)), only manageable by heavy parallelization. In contrast, feature extraction in the shallow case (n-gram features), only took 4 minutes.

Table 22 shows the performance of the Mintz feature set. Table 23 shows its performance when sentences longer than 50 words are skipped, which resulted in a 4-fold speedup in the parsing step. The best overall F1-score (*anydoc*) in the Mintz setting is 29.13%, which is substantially less than the 35.46% of the shallow standard feature set

j-parameters	Recall	Prec	F1	Recall *any*	Prec *any*	F1 *any*
0.1	0.0524	0.4375	0.0936	0.0753	0.6250	0.1345
1	0.1280	0.3782	0.1913	0.1850	0.5432	0.2760
10	0.1362	0.3210	0.1912	0.2111	0.4943	0.2958
tuned	0.1328	0.3714	0.1956	0.1980	0.5504	0.2913
cf. skip-grams, tuned	0.2350	0.3450	0.2795	0.2988	0.4360	0.3546

Table 22: **Mintz features, using the same sentences as in the standard pipeline.** Results of skip-n-gram feature set repeated for comparison. (*"any"* stands for the provenance-independent, more robust *anydoc* evaluation in the TAC scorer.)

j-parameters	Recall	Prec	F1	Recall *any*	Prec *any*	F1 *any*
0.1	0.0517	0.4523	0.0929	0.0719	0.6250	0.1290
1	0.1239	0.3791	0.1868	0.1802	0.5479	0.2712
10	0.1321	0.3227	0.1875	0.2049	0.4975	0.2902
tuned	0.1301	0.3797	0.1938	0.1905	0.5526	0.2833

Table 23: **Mintz features, not using sentences that exceed a maximum length of 50.** This greatly speeds up the parsing step.

(Table 20). Clearly, the features in the Mintz setting are strong in precision, but weak in recall, and it seems difficult to get a good balance between precision and recall, even when changing the j-parameter.

5.7.4. Aggregate vs. Single Sentence Training

Some work on distant supervision, such as Mintz et al. (2009), create their training data by making one instance for each argument pair, aggregating (i.e. adding up) the feature vectors obtained from the matching sentences. This is the approach we use in our standard pipeline, and we refer to it by the term *aggregate training*. Other approaches, such as Min et al. (2012a); Surdeanu et al. (2012) use every matching sentence as a single training instance, labeled as positive according to the distant supervision

j-parameters	Recall	Prec	F1	Recall *any*	Prec *any*	F1 *any*
0.1	0.0647	0.3754	0.1104	0.0911	0.5256	0.1553
1	0.1777	0.2034	0.1897	0.3187	0.3624	0.3391
10	0.2125	0.1138	0.1482	0.4132	0.2199	0.2871
tuned	0.1743	0.2803	0.2150	0.2954	0.4720	0.3634
cf. tuned aggregate	0.2350	0.3450	0.2795	0.2988	0.4360	0.3546

Table 24: **Single sentence training.**

assumption. We call this approach *single sentence training*. In this section, we compare aggregate and single sentence training with respect both to training runtime and to prediction performance.

The single sentence data contains 1959736 training instances, of which the sentences are grouped into 359839 training instances for the aggregate training. I.e. there are roughly 5 times as many instances for training in the single sentence setting. When measuring the CPU training time (without I/O), it turns out that single sentence training takes 481 hours (!), while aggregate training takes only 23 hours. The 20-fold increase in required training time (for only a 5-fold increase of instances) turns single sentence training impractical. In Joachims (1999) the authors report an empirical complexity for *SvmLight* of roughly $O(n^{2.0})$ with n the number of training instances, which fits our observation of super-linear growth.

Table 24 shows the end-to-end scores of single sentence training. The scores of single sentence training are comparable to the aggregate setting: slightly higher for the tuned *any*-doc setting, and lower for the exact setting and for the untuned *anydoc* settings (see Figure 20 on page 81). The exact setting is less comparable between runs, and the comparison to the tuned aggregate training should be done in the *any*-doc setting, since the aggregate results were part of the TAC 2013 submission and hence have been included in pooling the evaluation key. We conclude that single sentence training results in similar prediction quality while requiring a substantially higher training time.

5.7.5. Prediction

While training is done on an aggregate level, prediction is done on each candidate sentence independently. The per-sentence prediction is necessary since in TAC KBP, the task is not to find pairs that *likely* belong into the knowledge base (e.g. by indirect correlations), but to find pairs that *justifiably* belong into the knowledge base (i.e. actual sentences must express the relations). An answer is returned if at least one candidate sentence with it is classified as *true*.[35]

5.7.6. Summary

In this section we explored in detail the design choices for setting up the distant supervision SVM classifier module. We showed that gathering distant supervision training data from a knowledge base like Freebase can give very good results. We also explored an alternative way of generating distant supervision training data in a two-step process from seed patterns. This approach doubled the performance compared with applying the seed patterns directly – it performed, however, slightly worse than using Freebase. We also discussed a greedy scheme for optimizing the global F1 score of a multi-relation classifier set, and showed that it is essential for achieving good performance with the Freebase data.

We compared our shallow feature set, based on skip n-grams, to a popular feature set based on dependency parses. Not only does the shallow feature set drastically reduce the runtime of the prediction pipeline, it is also superior in terms of predictive power. Furthermore, we showed that training can be sped up without loss of accuracy by aggregating instances per entity pair.

5.8. Distant Supervision Patterns and Noise Reduction

As a second distant supervision component besides the SVM classifiers, we include scored plain surface patterns (the lexical token sequence between the arguments). The patterns are scored according to frequency in the distant supervision data, and by combining two noise reduction methods to suppress the influence of false positive matches.

[35]For single slot relations, only the answer with the highest classifier regression score is returned.

The pattern scoring follows the method described in Chapter 4.3, combining a generative topic model and a discriminatively trained perceptron for reducing the noise introduced by false positive distant supervision matches.

The overall scoring function used is the linear interpolation of the feature based topic model and the frequency-weighted perceptron score:

$$0.5 \cdot \frac{n(pat, topic(r))}{n(pat)} + 0.5 \cdot \frac{n(pat, r)}{n(pat)} \cdot \frac{P(r|pat, \theta)}{P(r|pat, \theta) + P(NIL|pat, \theta)}$$

The left term is the fraction in the training data that this pattern was assigned the respective relational topic (and not the pair-specific or background topic) by the feature based topic model. The right term is the feature-based perceptron score that decides whether the distant supervision examples underlying the pattern are rather to be treated rather as a true positive matches (predicting the label r) or as a false positive matches (predicting the label NIL), weighted by the relative frequency that this pattern was observed for that relation in the training data.

We denote the count of the pattern *pat* and the topic of relation r by $n(pat, topic(r))$, other counts are analogously denoted by $n(\cdot)$, and the feature-based perceptron probabilities by $P(\cdot|pat, \theta)$.

The scoring function provides scores in the interval between 0.0 and 1.0. We use the same training data as for the distant supervision SVM classifiers and use the global parameter tuning method to find score thresholds on the intertext patterns (see Algorithm 2 on page 84). We tune thresholds on the score levels $0.1, 0.3, 0.5, 0.7$ and 0.9.

5.9. Alternate Names Prediction

Slot fillers for the relation `alternate names` can be predicted by any of the validation components such as the SVM classifier or a pattern matcher. Additionally, we include a dedicated component that explicitly returns a slot filler for `per:alternate_names` or `org:alternate_names` if an expression returned by our query expansion (see Chapter 5.3.1) matches in one of the retrieved documents.

5.10. Post-processing and Redundancy Removal

Prediction scores are assigned to the responses that are judged positive by the SVM classifiers (Chapter 5.7) and by the distant supervision pattern matcher (Chapter 5.8).[36] For single-slot relations only the highest ranked slot filler is kept, ties are broken according to precedence in the retrieval step.

For list-valued relations, all positive responses are mapped to a normal form, based on Wikipedia link anchor text.[37] For every slot filler the top-1 expansion is calculated (as described in Chapter 5.3.1), which is in turn lower-cased and stripped off all non-letters and non-decimals. If two slot fillers are mapped to the same normal form, only the higher ranked slot filler is kept. Following the example of Figure 18 on page 65, if the surface forms *"Direction des Constructions Navales"* and *"DCNS"* were slot fillers for a relation/query combination, they both would be mapped to the normalized form *"dcns"*, and accordingly only one of them would be returned as an answer. Dates are normalized by a rule-based heuristic.

An optional step of post-processing is a relation-specific cut-off for the number of highest ranked answers returned per slot. While setting such thresholds on the development data (TAC KBP 2011 queries) improved performance, the 2012 runs (Roth et al., 2012) indicated that it did not have the expected positive effect. In the reported experiments, we do not use any cut-off on the number of returned answers per slot.

Additionally for the 2013 runs, due to an additional requirement in the task description for the `per:title` relation, we included job titles multiple times if they co-occurred with different organization names, and the co-occurrence was licensed by a pattern. [38]

[36] In the case of the SVM, the regression scores for one slot are normalized to lie between 0 and 1. The hand-crafted pattern matcher assigns a score of 1.0 to its matches. Slots returned by the alternate names component based on alias expansion get assigned a score of 0.5, in order to rank it lower than answers that have good contextual evidence.

[37] An exception is made for *org:alternate_names* and *per:alternate_names*, as here one is not interested in unique slot-filler entities but surface forms.

[38] The list of patterns was compiled from high-frequency context patterns between entities of type [PERSON] and [ORGANIZATION].

5.11. Non-Standard Modules

PRIS Syntactic Patterns. We implemented a module to match the dependency patterns provided by the PRIS team (Li et al., 2011). Thus we wanted to test whether dependency patterns may help to improve performance in our pipeline. Due to the many degrees of freedom to incorporate those patterns into a relation extraction system, we cannot guarantee that our module makes the best use of the provided patterns.
Wikipedia-Based Validator. This module runs the relation extraction pipeline on an additional Wikipedia text dump and uses the slot fillers thus obtained to validate candidates retrieved from the TAC corpora.

5.12. Single Component Analysis and Ablation Analysis

Component	P_{single}	P_{merge}	R_{single}	R_{merge}	$F1_{single}$	$F1_{merge}$
Alternate names	54.2	–	1.8	–	3.4	–
Seed patterns	50.2	50.4	10.3	12.0	17.1	19.4
Distsup Patterns	42.7	53.5	15.6	21.9	22.9	31.0
PRIS syntactic patterns	39.0	50.4	9.6	25.6	15.4	34.0
Distsup SVM classifier	34.7	40.5	23.6	34.3	28.1	37.2
Wiki validator	20.8	36.9	8.1	36.7	11.7	36.8
+inferred `per:title` affiliations	–	36.0	–	37.7	–	36.8
+relaxed query expansion	–	35.1	–	37.8	–	36.4

Table 25: **Performance of single component and merged component responses.** Components are sorted by precision. The last two components cannot be evaluated in isolation: the component "inferred `per:title` affiliations" operates on an already existing response, the component "relaxed query expansion" influences the number of candidates fed into validation components. These two components are evaluated in conjunction with the merger of all components up to the Wikipedia-based validator.

Table 25 shows the performance of the single components and the merger of their responses. See also Appendix A.3 on page 134 for a relation-specific breakdown of the

Component	P	R	F1	F1 gain
main run	42.5	33.2	37.3	
−Query expansion	41.1	17.5	24.5	+12.8
−Distsup SVM classifier	53.3	21.8	30.9	+6.4
−Distsup patterns	39.6	28.6	33.2	+4.1
−Seed patterns	38.2	29.5	33.2	+4.1
−Alternate names	41.1	31.0	35.4	+1.9
−Redundancy removal	41.4	33.2	36.8	+0.5
−Multiple `per:titles`	44.0	33.0	37.7	−0.4

Table 26: Precision, Recall and F1-score of the main run configuration when removing single components (one at a time), as well as the F1 gain contributed by the respective component on top of the other components. Components are sorted by complementary F1 gain.

single component performances.

In order to show how complementary those components are to each other, Table 26 gives an ablation analysis on the best-performing run (*lsv1*). The ablation analysis evaluates the complete system with single components deactivated one at a time.

Some observations on the performance of single components:

- **Alternate Names.** The inferred `alternate_names` slot fillers from the query expansion are of high precision. Although concerned with only two relations, this component gives an F1 gain of 1.9% on top of the other components.

- The **seed patterns** provide high-precision responses, but have relatively low recall for a component modeling all relations. They are considerably complementary to the other components (+4.1% F1).

- **Distsup patterns.** The patterns induced from the distant supervision data provide good-precision responses with good recall. They capture information not modeled by either the SVM classifiers or the seed patterns (+4.1% complementary F1 gain).

- **PRIS syntactic patterns.** The dependency patterns show good precision, but are slightly behind plain surface patterns in our experiments. One reason that the manual syntactic patterns of the PRIS system have less precision than plain surface patterns might be that it is difficult to intuitively write dependency patterns, and that syntax-based representations might in general have shortcomings in capturing relational information (see also Chapters 5.7.3 and 5.13).

- **Distsup SVM classifier.** The SVM classifiers are the strongest relation validation component in our system, both in terms of single performance as well as in complementary F1 gain (+6.4% F1).

- **Wikipedia-based validator.** This is the component with the lowest precision, since apart from candidate generation (query matching, tagging) only overlap with answers from Wikipedia is checked. It is interesting to note that while this component obtained high *anydoc* precision in our internal development benchmarks on the 2012 data, precision was rather low for the official run submitted.

- **Inferred `per:title` affiliations.** Inferring `per:employee_or_member_of` from predicted `per:title` relations had a minimal effect on the precision/recall ratio.

- **Query expansion** and **Relaxed query expansion.** It is important to note that query expansion has a high effect on overall performance, contributing a F1 gain of 12.8%. This is due to the greatly positive effect on recall, while exhibiting only a slightly negative impact on precision. Query expansion plays a role in both document retrieval and query matching. It seems necessary not to over-generate, as predicting more ambiguous aliases (no link-back requirement, see Chapter 5.3.1) increased recall but had negative effect on F1-score.

- **Redundancy removal.** Removing redundant slot fillers using Wikipedia anchor text had a slightly beneficial effect on overall F1.

- **Multiple `per:titles`.** On the other hand, trying to cluster predicted `per:titles` by their affiliations (see Chapter 5.10) was detrimental to performance.[39]

[39] The components related to post-processing, *redundancy removal* and *multiple `per:titles`* are part of every run in Table 25 and therefore only separately evaluated in the ablation study (Table 26).

5.13. Discussion: Shallow vs. Deep Analysis

In the standard configuration of *RelationFactory*, no deep linguistic analysis, such as dependency parsing, is used. Merely named-entity tagging is used to identify slot filler candidates – all features and patterns operate directly on the surface level. When developing the *RelationFactory* KBP system, we kept experimenting with more linguistically motivated representations but found that they did not provide any (substantial) gain compared with representations derived directly from the surface forms. While syntactic structures (especially dependency relations) are popular choices for representing semantic relations, our observations suggest that taking one step back from the dependency view may clear the sight to more central aspects of certain information extraction tasks.

Apart from purely practical advantages of a shallow approach (e.g. faster code that is easier to maintain, applicability in low-resource settings), there are also more considerations:

- **Contextual cues.** Words or word sequences that do not express the relation but provide topical information and may disambiguate a relational expression are naturally included in a shallow feature representation. A dependency analysis, however, aims at stripping off those cues.

- **Micro-structures without content words.** Chan and Roth (2011) observe that in ACE 80% of the mention pairs in a relation do fall in a pattern type where the relation is not explicitly expressed by a content word. The four pattern types they identify are *Premodifier* (e.g. [the [Seattle] Zoo]), *Possessive* (e.g. [[California's] Governor]), *Preposition* (e.g. [officials] in [California]) and *Formulaic* (e.g. [Medford], [Massachusetts]).

- **Parsing errors.** While syntactic parses may be accurate for short distance dependencies, which also can be easily captured by surface patterns, for longer distances the dependency accuracy significantly decreases (McDonald and Nivre, 2007).

5.14. Summary

In this chapter we described the practical challenges for query-driven relation extraction from large corpora of text and the layout of the end-to-end system developed in this dissertation. We specified the interaction of all modules involved, and evaluated their impact on end-to-end performance, as well as how different design choices impact their performance in isolation. We highlighted the vital role of argument tagging and query expansion based on link anchor text, and we gave a detailed breakdown of the recall still missing in the system. We discussed pattern- and classifier-based prediction modules, and scrutinized the design choices regarding training data, feature representation and parameter tuning. The results suggest that a tuned SVM using a shallow feature set, in combination with distant supervision data from a knowledge base like Freebase, is an extremely competitive classification module in itself. A detailed ablation analysis shows that all discussed components contribute to the overall good performance of the system.

6. Sequence Labeling: An Alternative or Enhancement to Classifier-based Prediction?

In this chapter we motivate why viewing relation prediction as a sequence labeling task may be a promising alternative and complementary addition to framing the problem as a per-instance classification task (as done by current slot-filling systems). The hope would be that a relation tagger, not being constrained to using argument candidates provided by previous steps in the pipeline, can correct mistakes and biases from earlier modules. A tagger would also have the freedom to incorporate and weigh evidence provided by the more traditional modules.

We show initial results where a CRF tagger achieves respectable performance (well above the TAC median system performance) despite its simplicity. However, in the simple form explored here, the tagger only profits slightly from incorporating the prediction of a distant supervision SVM.

We conclude the discussion on relation tagging by drawing parallels to recent work on sequence labeling with *global constraints*, that could also be beneficial in another unsolved problem in relation prediction, namely the problem of *combining* of several relation prediction systems or modules.

6.1. Motivation

Figure 22 shows the *potential* gains in absolute terms of the end-to-end system developed in this work. Since the system is biased towards higher precision, more potential gains are possible on the recall side (Figure 22, left). However, as discussed in Chapter 5.4, this cannot simply be achieved by tuning some threshold parameter in the classifier, since many potential answers are lost in previous stages of the pipeline. We illustrate this in Figure 22 (right), summarizing the detailed breakdown of recall errors from Chapter 5.4.

Document retrieval, *query matching* and *coreference* errors would have to be tackled by dedicated algorithms. The interesting error class for the suggested extension to relation prediction is the large amount of *named entity* errors: In more than 30% of the cases the sought relation argument is not detected by the tagger at all, or a label sequence overlaps it only partially. Moreover, the classifier is trained on distant

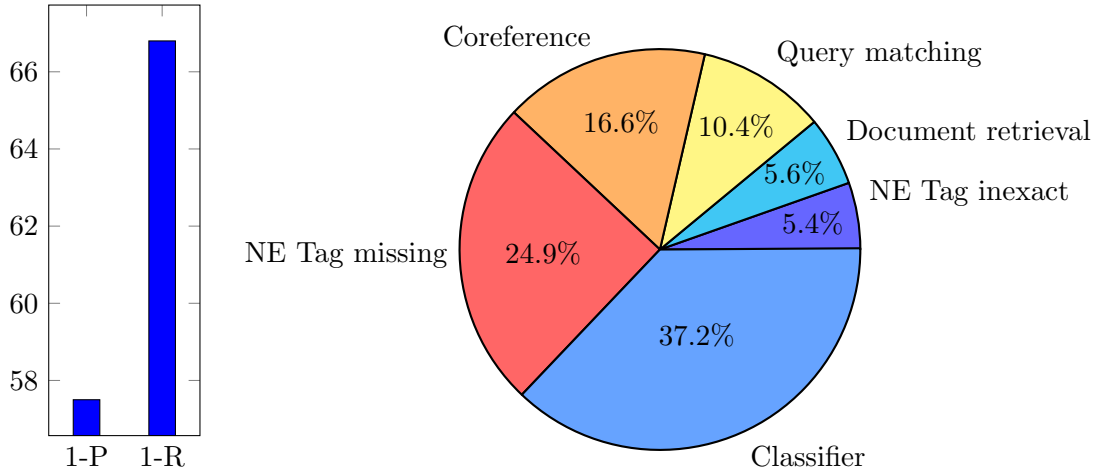


Figure 22: *Potential* of future relation extraction systems, relative to the methods developed in this work. **Left:** End-to-end precision and recall potential. **Right:** Recall potential, broken down into error classes.

supervision training data, which has certain biases (e.g. false positive errors) that are difficult to overcome and may negatively influence both precision and recall at the classification stage. A viable approach to overcoming the distant supervision biases is to tune the classifier thresholds (using a small amount of hand-annotated development data, see Chapter 5.7.2) and to keep high-precision predictions only. However, such an approach negatively impacts recall.

A relation argument tagger, trained on the development data, and with access to the output of the previous named entity tagging and SVM relation prediction modules, would have the potential to correct errors that would otherwise propagate from both stages, as it has the freedom to tag any sub-sequence as relational arguments. In particular, the challenge would be to set up a sequence labeling method for correcting

- systematic biases in the distant supervision SVM.
- systematic omissions by the tagger.

Both can be approached by training a tagger on the development data (the manually annotated key from previous TAC benchmarks), since this data is not constrained or influenced by biases from either the tagging or distant supervision prediction methods.

tokens	NE-tags	DS-SVM	label
[Q]	ORG	Q	Q
,		C	C
based		C	C
in		C	C
Mumbai	CITY	S	S
,			O
is			O
an			O
Asian	NATION		O
leader	TITLE		O
in			O
aluminum			O
and			O
copper			O
production	TITLE		O

tokens	NE-tags	DS-SVM	label
Livni	PERSON		O
is			O
also			O
due			O
to			O
travel			O
to			O
Boston	CITY		S
to			C
address			C
a			C
special			C
session			C
of			C
the			C
[Q]	ORG		Q

Figure 23: **Two training sentences for sequence labeling.** The shaded area marks the fields used in the feature template for the label of *Mumbay*.

6.2. Experiments

In a pilot experiment, we set up training data for a conditional random fields (CRF) tagger. We used *CRFsuite*[40], a state-of-the art sequence labeler, with the L-BFGS optimization (Nocedal, 1980) and standard settings for training and prediction. For training, we use the annotated keys from TAC 2009-2011, and align them with the tokenized source documents (when the query is matching and the character offsets of our tokenization agree with the key). This yields 4652 positive annotated sentences (containing a valid slot filler), or roughly 110 sentences per relation. This is considerably less (only 0.24% the amount) than the 1959736 positive training sentences (984325 per

[40] `http://www.chokkan.org/software/crfsuite/`

relation) used in distant supervision training. Furthermore, for the tagging training data, 323801 *negative* sentences are included, containing only a query match, but no valid slot filler.

Figure 23 illustrates the training data. Each sentence is represented by 3 feature columns: First, the tokens of the sentence. In order to avoid over-fitting to the queries used in training, the query tokens are wild-carded to *[Q]*. Second, the named entity tag column, containing the tags from the sequence tagger and the list matcher as described in Chapter 5.3.4. Third, the prediction of the Distant supervision SVM classifier (DS-SVM). This is included as a signal that can be used by the relation tagger. If the query and slot filler candidate pass the candidate generation stage, and the SVM classifies the instance as *true*, this column contains Q for each query token, S for each slot token, and C (context) for each token between query and slot. If the sentence is not recognized as candidate, or the SVM classifies it as *false*, no tokens are marked as DS-SVM features. The labels which are to be predicted are encoded similarly to the DS-SVM features: the label for the slot to be predicted is S; at prediction time, all maximal spans with that label are returned as answers. To mark the context that indicates a relation (and which may extend to sequences longer than what is covered by the template window as explained below), the label C is used. The label Q is used to mark the other boundary of relational context.

The CRF models two types of dependency: dependencies between a label and features, and dependencies between consecutive labels. For each label, features in a window of two preceding and two succeeding tokens are modeled. The CRF models *factors* (log-linear potential functions) between the label and all uni-grams in the feature window, and between the label and all bi-grams of the *token* and *NE-tag* columns of the feature window. Another factor models the bi-gram dependence on the preceding label.

Table 27 shows the performance (TAC 2012 test set) of two tagging models as compared with the distant supervision SVM classifier. The first uses only token features and NE-tag features. The performance of 22.9% is remarkable, as it only uses 0.24% of the amount of training data as compared with the distant supervision SVM. The performance of this simple tagger is also considerably higher than the median team

score in TAC KBP 2012 (10% F1)[41] and the median team score in TAC KBP 2013 (16.1% F1) (Surdeanu, 2013). The second CRF (the full model as described above) aims at incorporating the signal from the distant supervision classifier as additional features. The improvement obtained by doing this is surprisingly small, and the overall score is even below the score of the distant supervision SVM alone.

predictor	recall	precision	F1
tagger (tuning data)	25.5%	20.9%	22.9%
tagger (tuning data + SVM signal)	28.1%	20.5%	23.7%
SVM classifier (tuned)	25.7%	33.0%	28.9%

Table 27: End-to-end performance on TAC 2012.

6.3. Discussion

These experiments allow for two conclusions: First, a relation tagger is a viable model for predicting relations, achieving good results even in a simple setup with very little training data. Second, the strategy for encoding information from additional modules as explicit sequential features is obviously not optimal (since at least reproducing the performance of the additional module could be hoped for).

The experiments raise the question of how to incorporate prediction signals in a more effective way, while preserving the freedom to agree only partially with them if other evidence is available. Research in this direction would also further progress in two other challenging related tasks: The combination of a series of prediction modules and the combination of several relation prediction systems. While the combination of different modules within the same system would be beneficial to a setup like the one developed in this work, the combination of several systems is studied as the task of *slot filler validation* and evaluated in a separate track in TAC KBP[42]. Slot filling validation is an especially hard task, and although at TAC KBP the validation systems had access to the prediction output of all 18 slot filling participants, small improvements of 2.2%

[41] see `http://www.nist.gov/tac/publications/2012/papers.html`

[42] see `http://www.nist.gov/tac/2013/KBP/SFValidation/index.html`

on the top-ranked system were only possible after injecting additional human-created rules and knowledge (Yu et al., 2014).

Since the discussed tasks and problems are both promising and challenging, we think it might be interesting for future research to study novel approaches to tagging for them. One particularly interesting approach is tagging with constraints on tag ordering, tag existence and global constraints by dual decomposition (Belanger et al., 2014). Here, global soft constraints apply to the whole sequence, and can favor e.g. certain long-range configurations within a sentence. In the relation tagging case, such constraints could be *"If a slot filler is present, a query match must be present"* or *"If the classifier predicted* `true`*, a slot-filler must be present"*. We believe that such constraints would be even more helpful in event argument prediction (Reschke et al., 2014), where multiple possible arguments can optionally be realized (e.g. the event roles *buyer*, *seller*, *goods*, *time* and *place* for a *selling* event), and the assumption (from classifier-based relation modeling) that exactly two arguments are connected by a fixed set of context representation cannot be made. In an event setting, beneficial constraints could be *if a* `seller` *is present, a* `goods` *argument must be present*, certain canonical orderings can also easily be imagined. What makes the global constraints in the style of Belanger et al. (2014) so interesting is that they are soft, and that large amounts of constraints can be generated semi-automatically, while the bad constraints are filtered out by the learning algorithm. Future research will have to show which kinds of constraints, if any, are beneficial to the types of problems discussed in this chapter.

6.4. Summary

In this chapter we showed that relational argument taggers are an interesting alternative to pipelined classification settings. Taggers have the freedom to override decisions of earlier steps in a pipeline, could incorporate diverse signals and can bridge the gap to modeling n-ary event relations with many (optional) arguments. Although the results of initial experiments do not reach the performance of the other methods developed in this dissertation, they are good enough (well over the median TAC score) to show the potential of such an approach.

7. TAC Run Characteristics and Comparison with Other End-to-End Systems

Chapter 5 described the different components available in the *RelationFactory* system, developed for TAC KBP. As the components add up to a certain complexity, it is interesting to compare how different setups and configurations perform when imposing constraints on them (e.g. no syntactic analysis) or aiming at specific objectives (e.g. high recall). Moreover, the combination of components should ideally not include components harmful to the overall result. In this chapter, we give a brief overview of the runs officially submitted to the TAC KBP 2013 benchmark (using the components from Chapter 5), and discuss the approaches chosen by other participants in the benchmark.

7.1. System Runs and Results

run id	run type	P	R	F1
lsv1	fast	42.5	33.2	37.3
lsv2	precision	50.9	25.9	34.3
lsv3	all	36.9	36.6	36.8
lsv4	recall	35.1	37.8	36.4
lsv5	all shallow	38.1	35.8	36.9
Stanford		28.4	35.9	31.7
NYU		16.7	53.8	25.6

Table 28: Official (exact) scores on 2013 runs submitted by team LSV, compared with the best submitted runs of the systems most similar in design (Stanford University, New York University).

Table 28 gives an overview of the Saarland University "Lehrstuhl für Sprach- und Signalverarbeitung" (LSV) runs submitted to TAC KBP 2013. They are characterized as follows:

- **lsv1 (Main Run):** In this run, only *fast* validation components are used, this means especially no syntactic analysis and no query-specific analysis of an addi-

tional Wikipedia dump. The fast components are the SVM classifier, the distant supervision patterns, the seed patterns, and the alternate names expansion module.

- **lsv2:** Only modules are included that produced *high precision* on the 2012 development data. This includes most components of *lsv1*, but not the SVM classifier. Additionally, the syntactic patterns are included in this run.

- **lsv3:** This contains *all validation components* with standard configuration. It includes all components from *lsv1* and *lsv2*, and the Wikipedia-based validator.

- **lsv4:** This is a *high-recall* run. In addition to the components of *lsv3*, the entity expansion is relaxed (ambiguous expansions are allowed), and `per:employee_or_member_of` slots are inferred from predicted `per:title` slots (if a title is predicted, then a co-occurring organization name may be returned).

- **lsv5:** This is a run that exclusively comprises *shallow* components (i.e. no syntactic analysis). It corresponds to *lsv1* together with the Wikipedia-based validator.

Interestingly, the *fast* run (*lsv1*), that only extracts surface-level features and matches linear patterns, is the best performing in terms of $F1$ score. Increasing the precision by concentrating on high-precision modules as well as increasing the recall by merging responses from more modules did not have an overall positive effect. It remains for future work to analyze whether additional improvements can be achieved by a more principled module combination scheme (rather than simply merging the responses).

The approaches most similar to ours are that of Stanford University and New York University. Both systems are based on a pipelined approach using a combination of distant supervision classifiers and rule-based prediction, the Stanford system also includes noise reduction in its distant supervision classifier. The rest of this chapter will give a detailed overview of the submission of these and other teams in TAC KBP 2014.

System/Team	Precision	Recall	F1
LSV (*RelationFactory*)	42.53	**33.17**	**37.28**
ARPANI	50.38	27.45	35.54
RPI-BLENDER	40.73	29.02	33.89
PRIS2013	38.87	27.59	32.27
BIT	61.35	21.73	32.09
Stanford	35.86	28.41	31.70
NYU	53.83	16.76	25.56
UWashington	**63.45**	10.29	17.70
CMU_ML	32.30	10.69	16.07
SAFT_KRes	15.67	14.99	15.32
UMass_IESL	10.88	18.46	13.69
utaustin	25.16	8.11	12.26
UNED	17.59	9.33	12.19
Compreno	9.74	12.74	11.04
TALP_UPC	7.69	9.81	8.62
IIRG	7.72	2.86	4.17
SINDI	7.84	2.59	3.89
CohenCMU	1.98	3.68	2.57
Human Control Annotators	85.60	57.08	68.49

Table 29: Performance of other participating systems in the tac 2013 evaluation, according to the overview paper (Surdeanu, 2013).

7.2. Overview of Other TAC KBP Systems

In the following, we will describe the 10 best performing systems in TAC KBP 2013 slot-filling, as far as the participants have published a system description paper. The teams *ARPANI* and *CMUML* have not provided papers summarizing their systems.

The *RPI-BLENDER* workshop paper (Yu et al., 2013) does not contain any information about the regular slot filling system, but only about the team's temporal slot filling and slot filling validation systems. We assume that the *RPI-BLENDER* slot filling system is based on the open source *BLENDER* system (Chen et al., 2010). However, considerable additional work must have been included, since two other systems (*SAFT_Kres* (Chalupsky, 2013) and *utaustin* (Bentor, 2013)) are equally based on

the *BLENDER* system but only achieve about half the performance. The *BLENDER* system contains the following modules:

- A distant supervision pattern extractor trained with seed facts from Wikipedia infoboxes.
- A supervised relation predictor trained on annotated ACE relations that are manually mapped to TAC KBP relations where possible.
- Relation-specific manual rules to filter out answers, e.g. based on dependency paths.
- Output of the *OpenEphyra* QA system (Schlaefer et al., 2007), based on 68 manually created question templates.
- Patterns based on regular expressions.
- Direct fact lookup from Freebase.[43]
- Answers obtained from different modules are re-ranked using a maximum-entropy re-ranker.
- Cross-slot reasoning on the returned answer to filter out contradicting answers (e.g. a person cannot be the *parent* and *child* of the same person).

The *PRIS2013* (Li et al., 2013) system is based on a manual pattern set that is iteratively grown by a bootstrapping process. Unfortunately, neither the performance of the initial manual pattern set is provided, nor are details of the applied algorithm described. The *BIT* system (Xu et al., 2013a), too, is based on dependency patterns: initial manually written patterns are semi-automatically expanded by replacing content words with synonyms. In total the system uses 20 000 patterns. Additionally, in one of the submitted runs, the *BIT* system uses a classifier based on lexical and syntactic features – unfortunately no information about the type of training data is provided. By focusing only on manually constructed dependency patterns, and allowing some additional variation only on the lexical level, the *BIT* system (without the classifier) achieves a remarkably high precision of 61.4%.

[43]This was allowed until the TAC 2012 benchmarks but not in 2013

The main component of the *Stanford* system (Angeli et al., 2013) is a MIMLRE (see section 3.2.1) classifier trained on distant supervision data obtained from matching manually mapped Freebase relations against TAC corpora from the years 2010 and 2013, and a Wikipedia dump. The negative data is generated by selecting argument pairs that would be in contradiction to the information from the database using compatibility constraints. A set of 13 manually defined re-writes and constraints (e.g. consistency between country and state of birth) is also used to post-process answers after prediction. Furthermore, 69 manually crafted regular expressions are used to predict additional relational instances. A further focus of the *Stanford* system is document retrieval, were queries are successively expanded in order to achieve a desired recall of at least 50 documents, and named entity tagging, where the Stanford NER tagger was augmented by 74 000 regular expression rewrite rules to increase recall on slot-candidate level.

The *NYU* slot filling system (Grishman, 2013) is the same as the *2nd*-ranked system in 2012, described in Min et al. (2012a). The system uses hand-written patterns (contributing slightly more than in our system, see section 5.6), patterns iteratively bootstrapped from the manual patterns and a distant supervision classifier. The distant supervision classifier matches entity pairs from manually mapped Freebase relations on the TAC corpora. The quality of the seed pairs is increased by a heuristic based on point-wise mutual information. The relational contexts where any of the seed pairs match, are then *relabeled* before training the classifier. Relabeling means to assign another label than the one obtained from the seed pairs. Two processes are at work for relabeling: First, a set of classifiers is trained on the initial distant supervision data, and then all sentences are relabeled by using that classifier. The idea behind this mechanism is to reduce overlap between different relations that might confuse training the final classifier. Second, if a manual pattern conflicts with a relational annotation from distant supervision, the relation of the pattern is used. A set of maximum entropy classifiers is then trained on the relabeled distant supervision data. Instead of tuning a cost parameter, they establish a constant ratio between positive and negative data by sampling a subset of the negative examples. There is some detailed per-component evaluation for the *NYU* system on the 2012 TAC KBP queries: The *NYU* system achieves distant supervision scores (*anydoc* evaluation) of 14.4% F1-score, which is

markedly below our distant supervision classifier scores for the 2012 queries (28.8% F1-score).

The *UWashington* system (Soderland et al., 2013) pre-processes the TAC corpus to find occurrences of relations from unsupervised relational clusters (Banko et al., 2007), obtained by the OpenIE system [44]. The clustered relational representations consist of dependency paths connecting the arguments and containing at least one content word (verb or noun). Patterns were manually written for the TAC KBP relations, and if one of the manual patterns was contained in a cluster, all patterns of the cluster were used to predict the respective TAC KBP relation. The performance of two manual pattern sets is reported for the *UWashington* system: A small pattern set, using 123 manual patterns (constructed in *3 hours* of work), and a second, larger, pattern set using 492 patterns that were continuously refined testing on the 2012 KBP answer key. Similar to the *BIT* system, the *UWashington* approach, focusing on dependency based representation and limiting variation, yields a remarkably high recall of 63.5% at the expense of recall.

The *SAFT_Kres* system (Chalupsky, 2013) combines the output of the *BLENDER* system with a rule-based reasoning system that operates on document level after syntactic parsing (Chalupsky, 2012). This system addresses only 13 relations that were most frequent in TAC 2012.

The *UMass_IESL* (Singh et al., 2013) system significantly differs from most other approaches: instead of applying a query-driven pipeline that applies relation-specific models, predictions are made for all relevant information in the corpus using Universal Schema (Riedel et al. (2013), see also Chapter 4.4). A matrix of co-occurrence counts is built with all entity pairs in the corpus as rows, and with surface patterns and TAC KBP relations as columns. A low-rank embedding of this matrix is computed, which can be used for similarity computation between entity pairs and TAC KBP relations. Since the matrix approximation is low-rank, it is forced to generalize and to express similarity between argument pairs and relations even if they do not co-occur in the training data. This way Universal Schema leverages soft and indirect associations between patterns, relations and entities. Additionally, hand-crafted rules are applied for around half the relations.

[44] The source code for this is available at: `https://github.com/knowitall/openie`

7.3. Summary

In this chapter we have compared the LSV *RelationFactory* slot filling system with other systems in TAC KBP 2013. In comparison with other systems, several characteristics stand out from our approach: the feature-set is chosen to be purely shallow (ngram- and skip-ngram-based), in contrast to most other systems, that work on dependency representations. Apart from the *Stanford* system, it is the only system to include a noise reduction algorithm for distant supervision. Effective query expansion, and reliance on learned classifiers (for which thresholds can be tuned) rather than high-precision patterns, result in a high recall compared with other systems.

8. Matching of Relational Arguments in the Food Domain

In Chapters 5.4 and 5.4.1, we showed that argument tagging is crucial for TAC KBP relations. Most relations in TAC KBP belong to standard named entity types such as *PERSON* or *ORGANIZATION* and are tagged with sequence labeling methods trained from manually annotated data (Chrupała and Klakow, 2010). For entity types where such training data does not exist (e.g. *CAUSE_OF_DEATH* or *CRIMINAL_CHARGES*) we resorted to context-insensitive matching of entities using lists compiled from Freebase.

In this chapter we study the problem of how the prediction of relations between entities of one *coarse-grained* type can be improved by predicting the *fine-grained* subtypes of the arguments. This problem arises when predicting relations within one *domain*, as it is the case for *products* of the same coarse-grained domain type that can be combined with one another depending on the particular fine-grained subtypes. Classes of *products* naturally provide closed domains with subtypes; they also are motivated by obvious (commercial) use-cases, such as product recommendation and information systems. We exemplify our approach by relation extraction in the *food* domain because it provides for an exceptionally rich domain of general interest. However, the general methodology should be applicable to other domains such as for example *fashion*.[45]

8.1. General Setup and Motivation

As will be shown in this chapter, relation extraction in the food domain not only depends on finding the food entities suitable to be potential relation arguments, but can also profit from a more fine-grained type-modeling. Finding all food entities on a coarse level is already a task that could not be solved by a standard named entity tagger, but only by specially tailored solutions, such as list matching methods as discussed in Chapter 5.3.4 (e.g. by taking lists of food entities from knowledge bases like Freebase). For a more fine-grained modeling of subtypes that goes beyond the granularity of such resources this is not possible anymore. In this chapter we will describe our semi-supervised approach to this task. This work is evaluated on the gold standard

[45]This work is part of a bigger research project in collaboration with Michael Wiegand, and this Chapter focuses on the main contribution by Benjamin Roth.

Relation	Description/Example	Freq.	Perc.
SuitsTo	food items that are typically consumed together	633	42.20
	My kids love fish fingers with mashed potatoes.		
SubstitutedBy	similar food items commonly consumed in the same situations	336	22.40
	We usually buy margarine instead of butter.		
IngredientOf	ingredient of a particular dish	246	16.40
	Falafel is made of chickpeas.		
Other	other relation *or* co-occurrence of food items are co-incidental	285	19.00
	On my shopping list, I've got bread, cauliflower, ...		

Table 30: Food relation types and their respective frequency on the gold dataset.

annotation on a German food corpus by Wiegand et al. (2012b). The particular relation extraction task is to detect instances of the relations *SuitsTo*, *SubstitutedBy* and *IngredientOf* as shown in Table 30.

These relations are highly relevant to customer advice and to recommendation for a wide range of consumer products, not only in the food domain, which is especially obvious for the relations *SuitsTo* and *SubstitutedBy*: Customers want to know which items can be used together (*SuitsTo*), be it two food items that can be used as a meal or two fashion items that can be worn together. Substitutes are also relevant to situations in which item A is out of stock but item B can be offered as an alternative.

For extracting potential relational arguments, we employ a list of 1 888 food items from Wiegand et al. (2012a) of which 1 104 items were directly extracted from GermaNet (Hamp and Feldweg, 1997), the German version of WordNet (Miller et al., 1990), and another 784 items manually added. The GermaNet items were identified by extracting all hyponyms of the synset *Nahrung* (English: *food*), the manual items were obtained by asking annotators for typical slot fillers for partially instantiated relations, i.e. relation instances for which only one of the two arguments was provided. For evaluating the performance of relation extraction, the partially instantiated relations are provided to the system, and the automatically found answers (the initially missing arguments) are compared with the gold-standard.

While candidate argument recall is not a problem in this setting since the list for

matching food items has been extended by manually adding missing food items, we will illustrate in the following that semi-automatic induction of more fine-grained argument *subtypes* are desirable for relation extraction performance. The desired slot-fillers of all relations considered in our experiments for the food domain are of type FOOD_ITEM, and therefore relations are easily confused by an automatic method since the argument type cannot serve as a disambiguator between relations. Contextual information may be used for disambiguation, but there may also be frequent contexts that are not sufficiently informative. For example, 25% of the instances of *IngredientOf* follow the lexical pattern *food_item*$_1$ *with food_item*$_2$ (Example 1). However, the same pattern also covers 15% of the instances of *SuitsTo* (Example 2).

1. We had a stew with red lentils. *(Relation: IngredientOf)*
2. We had salmon with broccoli. *(Relation: SuitsTo)*

More fine-grained food types may give additional cues to the system, for example as to which of the food items are dishes. Only in 1, there is a dish, i.e. *stew*. So, one may infer that the presence of dishes is indicative of *IngredientOf* rather than of *SuitsTo*.

food_item$_1$ *and food_item*$_2$ is another ambiguous context. It can be observed not only with the relation *SuitsTo*, as in 1 (66% of all instantiations of that pattern), but also with *SubstitutedBy* (20% of all mentions of that relation match that pattern), as in 2. For *SuitsTo*, the food items often belong to pairs of characteristic classes of food, for example *meat* is commonly served with a *starch*-based side dish or *vegetables*. For *SubstitutedBy*, the two food items are very often of the same category.

1. I very often eat fish and chips. *(Relation: SuitsTo)*
2. For these types of dishes you can offer both Burgundy wine and Champagne. *(Relation: SubstitutedBy)*

Note that the relation types *SuitsTo* and *SubstitutedBy* connect items within the same domain but are not specific to the food domain. Therefore, type-based disambiguation between those relations may also be relevant to other life-style domains.

8.2. Methodology

We achieve a better type modeling by inducing a clustering for food item sub-categorization from very little seeds. In particular, we are interested in two types of categorization:

Class	Description	Size	Perc.
MEAT	meat and fish (products)	394	20.87
BEVERAGE	beverages (incl. alcoholic drinks)	298	15.78
VEGE	vegetables (incl. salads)	231	12.24
SWEET	sweets, pastries and snack mixes	228	12.08
SPICE	spices and sauces	216	11.44
STARCH	starch-based side dishes	185	9.80
MILK	milk products	104	5.51
FRUIT	fruits	94	4.98
GRAIN	grains, nuts and seeds	77	4.08
FAT	fat	41	2.18
EGG	eggs	20	1.06

Table 31: The different food types (*gold standard*).

1. the 11-class partitioning according to Food Guide Pyramid (U.S. Department of Agriculture, 1992) (Table 31)

2. a binary partitioning into dish vs. non-dish food.

We compare an unsupervised and a semi-supervised graph clustering approach on a co-occurrence based graph built from the corpus.

In the semi-supervised case, as little as 10 prototypical seeds for each of the 11 Food Guide Pyramid classes and 100 seeds for each of the two classes dish/non-dish are used. The graph-based methods are compared with two baselines: (1) a suffix-based heuristic, where classes are assigned according to partial matches of the surface form with seed items; (2) a lookup of food items according to the categories in GermaNet that roughly correspond to the desired food types. Furthermore, we show, that the obtained food categorization helps in predicting relationships between food-items.

8.3. Building the Food Graph

To enable a graph-based induction, we generate a similarity graph that connects similar food items. For that purpose, a list of domain-independent similarity-patterns was

compiled. Each pattern is a lexical sequence that connects the mention of two food items (Table 32). Each pair of food items observed with any of those patterns is connected via an edge in the graph, weighted by the count of all pattern matches for the pair (the different patterns are treated equally).

Due to the high precision of our patterns, with one or a few prototypical seeds we cannot expect that all items of interest are directly connected by the patterns to the seed items of the correct food category. Instead, one also needs to consider transitive connectedness within the graph. For example, in Figure 24 *banana* and *redberry* are not directly connected but they can be reached via *pear* or *raspberry*. However, by considering intermediate relationships it becomes more difficult to determine the most appropriate category for each food item: most food items are indirectly connected to food items of several different categories (in Figure 24, there are not only edges between *banana* and other types of fruits but there is also some edge to some sweet, i.e. *chocolate*).

For the final class assignment and disambiguation, we apply a robust graph-based clustering algorithm (Belkin and Niyogi, 2004). For the items in the example graph, it will figure out that *banana*, *pear*, *raspberry* and *redberry* belong to the same category and *chocolate* belongs to another category, since it is mostly linked to many other food items not being fruits.

Patterns	food_item$_1$ (*or*|*or rather*|*instead of*| *"("*) food_item$_2$
Example	{*apple*: *pineapple*, *pear*, *fruit*, *strawberry*, *kiwi*}
	{*steak*: *schnitzel*, *sausage*, *roast*, *meat loaf*, *cutlet*}

Table 32: *Domain-independent* patterns for building the similarity graph.

8.4. Semi-supervised Graph Clustering

For semi-supervised graph optimization we apply the label prediction method described in Belkin and Niyogi (2004), a robust algorithm that only contains few free parameters to adjust. It is based on two principles: First, similar data points should be assigned similar labels, as expressed by a similarity graph of labeled and unlabeled data. Second, for labeled data points the prediction of the learnt classifier should be consistent with

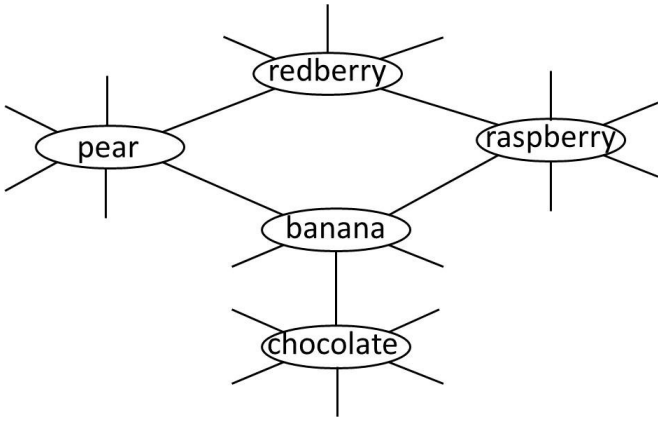


Figure 24: Illustration of the similarity graph. Two nodes are connected in the graph by a link if they are connected in the text by one of the domain-independent patterns.

the (actual) gold labels. In many scenarios, a graph is constructed in such a way that a weighted edge exists between two vertices if the similarity between the corresponding data points exceeds a specified threshold, or one data-point is a k-nearest-neighbor of the other. In our case, textual co-occurrence between food items (as encoded by our similarity graph) defines a straightforward method to set both edges and weights.

We construct a weighted transition matrix W of the graph by normalization of the matrix with co-occurrence counts C which we obtain from the similarity graph (Chapter 8.3). We use the common normalization [46] by a power of the degree function $d_i = \sum_j C_{ij}$: it defines $W_{ij} = \frac{C_{ij}}{d_i^\lambda d_j^\lambda}$ if $i \neq j$, and $W_{ii} = 0$. The normalization weight λ is the first of two parameters used in our experiments for semi-supervised graph optimization. For learning the semi-supervised classifier, we use the method of Zhou et al. (2005) to find a classifying function which is sufficiently smooth with respect to both the structure of unlabeled and labeled points.

Given a set of data points $\mathcal{X} = \{x_1, \ldots, x_n\}$ and a label set $\mathcal{L} = \{1, \ldots, c\}$, with $x_{i:1 \leq i \leq l}$ labeled as $y_i \in \mathcal{L}$ and $x_{i:l+1 \leq i \leq n}$ unlabeled. For prediction, a vectorial function $F : \mathcal{X} \rightarrow \mathbb{R}^c$ is estimated assigning a vector F_i of label scores to every x_i. The predicted labeling follows from these scores as $\hat{y}_i = \arg\max_{j \leq c} F_{ij}$. Conversely, the gold labeling

[46] see e.g. `http://www.ml.uni-saarland.de/GraphDemo/GraphDemo.html`

matrix Y is an $n \times c$ matrix with $Y_{ij} = 1$ if x_i is labeled as $y_i = j$ and $Y_{ij} = 0$ otherwise.

Minimizing the cost function $\mathcal{Q}$ aims at a trade-off between information from neighbors and initial labeling information, controlled by parameter μ (the second parameter used in our experiments):

$$\mathcal{Q} = \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{ij}\left\|\frac{1}{\sqrt{\delta_i}}F_i - \frac{1}{\sqrt{\delta_j}}F_j\right\| + \mu\sum_{i=1}^{n}\|F_i - Y_i\|\right)$$

where δ_i is the degree function of W.

The first term in $\mathcal{Q}$ is the smoothness constraint; its minimization leads to adjacent edges having similar labels. The second term is the fitting constraint; its minimization leads to consistency of the function F with the labeling of the data. The solution to the above cost function can be efficiently found by solving a system of linear equations (Zhou et al., 2005).

As we do not possess development data for this work, we set the two free parameters $\lambda = 0.5$ and $\mu = 0.01$. This setting is used for both induction tasks and all configurations. It is a setting that provided reasonable results without any notable bias for any particular configuration we examined.

8.5. Unsupervised Graph Clustering

We also examine an *unsupervised* method (**UNSUP**) that applies spectral clustering on the similarity graph following the algorithm described in Von Luxburg (2007):

- Input: a similarity matrix W and a number k of categories to detect.[47]
- The laplacian L is constructed from W. It is the symmetric laplacian $L = I - D^{1/2}WD^{1/2}$, where D is a diagonal degree matrix.[48]
- A matrix $U \in \mathbb{R}^{n \times k}$ is constructed that contains as columns the first k eigenvectors $u_1, \ldots, u_k$ of L.

[47] W is used as in the semi-supervised experiments (see Chapter 8.4), but without normalization. k corresponds to the number of categories to detect, that is, 11 for the *food type* categorization and 2 for the detection of *dishes*.

[48] That is, D_{ii} equals to the sum of the ith row.

- The rows of U are interpreted as the new data points. The final clustering is obtained by k-means clustering of the rows of U.

UNSUP (which is completely parameter-free) gives some indication of the intrinsic expressiveness of the similarity graph as it lacks any guidance towards the categories to be predicted.

8.6. Experiments: Type Clustering

We evaluate two types of clustering: the semi-supervised graph clustering (**SEMISUP**, see Chapter 8.4) using the manual seeds, and the unsupervised one (**UNSUP**) according to the method described in Chapter 8.5. We include two baselines: The first induces a categorization of food items according to overlap with the manually selected suffixes (**HEUR**), the second uses manually mapped GermaNet synsets instead of the seeds (**GermaNet**).

HEUR rests on the observation that German food items of the same food category often share the same suffix, e.g. *Schokoladen<u>kuchen</u>* (English: *chocolate cake*) and *Apfel<u>kuchen</u>* (English: *apple pie*). For HEUR, we manually compiled a set of a few typical suffixes for each food type/dish category (ranging from 3 to 8 suffixes per category). For classification of a food item, we assign the food item the category label whose suffix matched with the food item.[49]

The **GermaNet** baseline makes use of the semantic relationships encoded in GermaNet. Our two types of food categorization schemes can be approximated by the hypernymy graph in that ontology: We manually identify nodes that resemble our food categories (e.g. *fruit*, *meat* or *dish*) and label any food item that is an immediate or a mediate hyponym of these nodes (e.g. *apple* for *fruit*) with the respective category label. The downside of this method is that a large amount of food items is missing from the GermaNet-database (see Chapter 8.1).

In graph-based food categorization, one can only make predictions for food items that are connected (be it directly or indirectly) to seed food items within the similarity

[49] Unlike German food items, English food items are often multi-word expressions. Therefore, we assume that for English, instead of analyzing suffixes the usage of the head of a multi-word expression (i.e. *chocolate <u>cake</u>*) would be an appropriate basis for a similar heuristic.

Configuration	Acc	Prec	Rec	F1
HEUR	25.5	**87.9**	32.2	42.9
GermaNet	75.4	73.2	75.0	72.4
UNSUP	56.1	41.0	42.5	38.4
SEMISUP	**80.2**	75.9	**80.6**	**77.7**

Table 33: Comparison of different classifiers for the Food Guide Pyramid categorization.

Configuration	Acc	Prec	Rec	F1
HEUR	74.1	**84.3**	59.9	58.6
GermaNet	79.0	75.9	75.5	75.7
UNSUP	67.9	59.0	50.0	40.6
SEMISUP	**83.0**	80.8	**79.5**	**80.1**

Table 34: Comparison of different classifiers distinguishing between dishes and elementary food items.

graph. For the GermaNet baseline, the coverage problem is even worse. To expand labels to unconnected and unclassified food items, we apply a post-processing heuristic similar to HEUR which exploits the suffix-similarity of food items. It assigns each unconnected food item the label of the food item (that could be labeled by the graph optimization) that shares the longest suffix. This heuristic is applied to both graph methods (UNSUP and SEMISUP) as well as the GermaNet method.[50]

Tables 33 and 34 show a comparison of the different methods to classify food items into categories. The suffix heuristic is fairly precise, but can only capture very few items for which an overlap with manually selected suffixes exist, and fails entirely otherwise. Better results (for all measures that include recall in the calculation) are achieved by using GermaNet. However, using this resource implicitly makes use of a labor-intensive human ontology construction effort. It is interesting to see that the unsupervised graph method, without using any human input, exhibits a degree of

[50]The post-processing heuristic is not applied on HEUR as it would produce no changes.

accuracy roughly equivalent to or better than the suffix heuristic.[51] The precision of the unsupervised method is low, however. The semi-supervised graph-based method, although making use of only 10 seeds per category in the food pyramid, outperforms GermaNet on all measures, and compares favorably to the surface form heuristic in all cases where recall is important. To summarize, one can see that the semi-supervised label propagation method is an effective way to partition food items into subcategories, and can even beat manually constructed resources such as GermaNet.

8.7. Experiments: Improving Relation Extraction by using Type Clusters

In this section, we will examine the effect of using the induced fine-grained type clustering in a relation extraction setting. For our experiments, we used a crawl of `chefkoch.de` (Wiegand et al., 2012b) consisting of 418 558 web pages of food-related forum entries. `chefkoch.de` is the largest German web portal for food and recipes.

We randomly extracted 1 500 sentences from the text corpus in which two (arbitrary) food items co-occur. To each sentence one of the relation types enumerated in Table 30 was manually assigned: *SuitsTo*, *SubstitutedBy*, *IngredientOf* and *Other* (for cases in which either another relation between the target food items is expressed or the co-occurrence is co-incidental). On a subset of 200 sentences, we measured a substantial inter-annotation agreement (Landis et al., 1977) of Cohen's $\kappa = 0.67$.

For predicting relations, we trained a multi-class SVM-classifier, using the features described in Table 36. Using the fine-grained food types alone, from either GermaNet or the graph method, can already establish a surprisingly good baseline. Word features capture much of the relational context and lead to a more robust predictor at 55.1% F1. Adding the types from the GermaNet ontology can lead to a modest improvement on that, whereas adding the semi-supervised types from the graph method leads to significant improvements on the word feature baseline. A similar effect can be observed with the bigger feature set: The graph method leads to consistent and significant improvements over and above already strong feature sets and contributes more valuable fine-grained types than the manually constructed GermaNet resource. In conclusion, the information we induced from our domain-specific corpus cannot be obtained by

[51] For evaluation purposes, a resulting unsupervised cluster is assigned the most frequent label which its items would have according to the gold standard.

Features	Acc	Prec	Rec	F1
GermaNet	45.3	41.3	37.2	37.3
graph	46.0	39.4	39.7	38.6
word	60.1	56.9	54.5	55.1
word+GermaNet	61.3	58.6	56.0	56.7
word+graph	62.9	59.2	57.6	58.1°
word+patt+brown+synt+pos+conj	61.7	59.0	57.8	58.2*
word+patt+brown+synt+pos+conj+GermaNet	63.1	60.2	58.6	59.1°
word+patt+brown+synt+pos+conj+graph	**64.7**	**62.1**	**60.3**	**60.9**$^{\circ\dagger}$

statistical significance testing (paired t-test): better than *word* * at $p < 0.1$/ $^{\circ}$ at $p < 0.05$; † better than *word+patt+brown+synt+pos+conj* at $p < 0.05$

Table 35: Comparison of various features (Table 36) for relation extraction.

Features	Description
patt	lexical surface patterns as in Wiegand et al. (2012a)
word	bag-of-words features: all words within the sentence
brown	features using Brown clustering: all features from *word* but words are replaced by induced clusters
pos	part-of-speech sequence between target food items; tags of preceding and following word
synt	syntactic parse tree: path from first target food item to second target food item
conj	feature conjunctions: (*patt*, *pos*, *synt*) $\times$ brown classes of target food items
graph	semantic food information induced by graph optimization
GermaNet	semantic food information derived from GermaNet

Table 36: Description of the feature set.

other NLP-features, including word-class induction methods such as Brown clustering.

8.8. Summary

Argument tagging is a vital step in any relation extraction setup. However, for many argument types there is no annotated data for training sequence labellers that could

detect potential relational arguments. Sometimes this can be circumvented by string matching, using lists of names that belong to a certain type according to a knowledge base like Freebase. However, such lists can be incomplete or the granularity of such lists can be too coarse.

In this chapter we have shown that granularity is in fact a problem for relation extraction in the food domain: often, identical surface forms change their meaning depending on the fine-grained subtype of a relational food argument. We mitigated this problem by a minimally supervised type induction scheme using graph-based label propagation. Using as little as 10 prototypical seeds per fine-grained type (with total of 11 types), we cluster 1 888 food terms and get higher accuracy by this method than by deriving the type via heuristics from a curated knowledge source like GermaNet. Moreover, we showed that using fine-grained type information indeed helps in predicting relations in the food domain and is complementary to unsupervised type clusters such as Brown classes.

9. Outlook: The Future of Relation Extraction Evaluation

In the following we motivate why developments in Knowledge Base Population *evaluation* may be an important factor for further progress in the field of relation extraction. We discuss certain shortcomings of current slot-filling evaluation campaigns in TAC. While TAC evaluations give a realistic picture of the participating systems through pooling, the annotated data is incomplete for evaluation outside of TAC, and re-usability is limited. We propose a feasible alternative annotation scheme that would preserve the character of a query-driven evaluation campaign, while ensuring re-usability.

The TAC KBP setup is designed to give a realistic assessment of the performance obtained by the systems participating in the evaluation campaigns. The evaluation setup measures how many of the correct slots are filled, i.e. it measures the usefulness of the system output for constructing a knowledge base. At the same time it is required that correct slots come from justification contexts that indeed express the relation.

In this latter requirement the setup is distinct from other approaches that take an existing knowledge base as the ground truth that has to be recovered from text. The TAC requirement for justifications makes the setting more well-defined: given the query, any expressed relational information is correct, even if it is not present in an existing knowledge base. Compared with knowledge bases, which are often severely incomplete (Min et al. (2013) report e.g. that for 93.8% of persons in Freebase the attribute `/people/person/place_of_birth` is missing), the TAC gold key only contains few false negatives for judged answers. At the same time also the number of false positives in the gold key is reduced, since not only the slot must be predicted correctly (which would suffice for gold keys automatically extracted from a knowledge base), but also the justification must be correct.

Overall recall volume is estimated by pooling the answers of all participants and adding the results from a time-limited manual search. Therefore, the recall volume may be underestimated, and the key may contain systematic biases introduced by the participating systems. Since this equally affects all participating systems, fair comparability between all systems of the official evaluation is established.

However, for systems or methods that were not included in the official evaluation, the official evaluation method is not appropriate anymore. It may be that the gold

key does not contain a correct slot filler that is contained in a response that was not part of the pool. Furthermore, a correct slot filler may even be in the gold key, but with a different provenance (document or token offsets), and therefore scored as incorrect. This is especially severe for answers to *single*-slot relations that are repeated frequently in the corpus, but for which no more than one context may be returned by each system. We suspect that this latter case is the most frequent mismatch in evaluating new answers; this mismatch can be remedied to some degree by evaluating in the *anydoc* setting.[52]

What would a setting for relation extraction look like that would not suffer from some of the aforementioned shortcomings?

One approach would be to annotate a small subset of text extensively for all relation instances expressed in it, independently of any system responses. This would be similar to the previous ACE campaigns, but with the TAC knowledge base relations instead of linguistically motivated ones. The advantage here would be the complete coverage of the expressed relations in the test set. The downside would be that the test set would be smaller while requiring a much greater human annotation effort: Many irrelevant sentences would have to be screened for relation instances – in the current pooling setting only the likely candidates are looked at. Another disadvantage would be that the setting would lack the query-based retrieval step, and therefore be less challenging and less motivated by a realistic search scenario.

A combination of the query-based scheme and a document annotation scheme might help to establish comparability with later developed systems. Such a scheme could look like the following: For every query, those documents are retained that contain most slot fillers for the query. Greedily, such documents are added until a certain ratio, e.g. 80%, of the slot fillers is covered. Those retained documents would undergo

[52] By evaluating only according to micro-average scores (every *answer* has equal influence), relations with many answers dominate the scores. In order to measure better how methods perform over a wide range of novel relations, macro-averaging (every *relation* has equal influence) would be an alternative evaluation scheme to consider. However, the task organizers emphasize that the influence of the more frequent relations is not too dominant:

> "[...] to reach 60% coverage of the evaluation data, a system would have to model 13 slots, and these include more complex relations such as per:charges. " Surdeanu (2013)

further manual annotation to find potential additional slot filler occurrences for the query. Restricting annotation of those documents to only the respective query would limit manual annotation effort, while ensuring full query-specific recall coverage on the selected document set. Systems would then be evaluated with respect to this set of query-specific documents (and query-specific relational information).[53] The new gold key, limited to this set of documents, would be complete and could be used for evaluating systems developed after the evaluation campaign.

In this context it is important to note that construction and evaluation of test sets have been an object of major research efforts in the document retrieval community, mostly in the context of the TREC campaigns (Cormack et al., 1998; Voorhees, 2000; Voorhees and Buckley, 2002; Voorhees, 2002; Buckley and Voorhees, 2004), including the study of reliability and re-usability of relevance judgments (Zobel, 1998; Sanderson and Zobel, 2005). For TAC KBP, this is work that for the most part still needs to be done in order to have a reliable evaluation standard that allows for robust comparison of newly developed systems.

To conclude, creating re-usable evaluation data sets is an important area that, if developed, could increase the usefulness of data sets and the effectiveness of resources assigned to data set creation. The current pooling-based approach works well for assessing participating systems but is potentially inadequate for judging relation extraction systems not included in the pool. We have suggested an evalution scheme that lets annotators focus on documents containing most answers of the pooled systems. A manual, exhaustive, query-specific annotation of those documents could then be used as gold standard with a well-defined complete recall, and could serve as the basis for assessing future systems.

[53] Participating systems would only have to include an additional response where redundant occurrences of predicted slot fillers would be retained and not removed, so as to allow comparison w.r.t. to the new key, which is now limited to a subset of the documents.

10. Conclusion

The present dissertation is a study of all elements of the end-to-end Knowledge Base Population problem, the task of finding information in large amounts of text and structuring it according to a pre-specified schema. One focus was laid on quantifying the overall effects of design decisions at all stages of the pipeline, and another on developing solutions for the identified most important problems involved in the retrieval and extraction of relational information. This development lead to the *RelationFactory* Knowledge Base Population system which was top-ranked in TAC KBP 2013.

The scientific problems associated with the query-driven relation extraction setting this research is committed to, can be assigned to two stages: The first is a recall-oriented stage that includes problems from the information retrieval domain, such as query entity expansion, document retrieval, named entity tagging and entity type modeling. The second stage, precision-oriented relation modeling, deals with predicting the actual relations given the retrieved candidate contexts.

In the recall-oriented stage, entity modeling, such as finding the type and all possible surface forms of an entity, is central. Entity types are essential for finding all correct occurrences of relations. They are also able to disambiguate between ambiguous contexts: In the study of relation extraction in the food domain we showed how to improve relation prediction using minimally supervised fine-grained type modeling based on spectral graph clustering.

In the relation modeling stage, one focus is laid on the representation of relational contexts. We study a shallow representation and compare it with Mintz-features, a state-of-the-art representation on dependency parse features. The shallow representation outperforms the syntactic one. This is an interesting result that allows for faster relation extraction systems and has potential applications in low-resource settings. This result is in line with other research that has shown problematic aspects of syntax-based representations, such as low parsing accuracy for long-range dependencies, the fact that a big fraction of relations is expressed by formulaic expressions without content words, as well as the inability of syntax-based representations to capture contextual and topical cues that do not lie on the dependency path.

A special challenge in Knowledge Base Population is that the creation of training data is very labor-intensive due to the great variability through which a relation can be

expressed and that needs to be covered by a relation model. We have approached this problem in a setting known as distant supervision, where contexts are retrieved for pairs between which a relation is known to hold. These contexts are then used as positive training data. We have experimented with pairs coming from the Freebase knowledge base (the classical setting) as well as with pairs obtained by matching manual seed patterns (a novel variation on the distant supervision scheme). Our experiments show that both are viable strategies to get relational predictors with good precision and recall. The Freebase pairs generally lead to a better overall performance than the seed patterns. The distant supervision model based on seed patterns on the other hand shows less sensitivity to parameter tuning and is a good alternative for settings where no initial knowledge base is available.

Since the distant supervision training data is created semi-automatically and heuristically, it is inherently noisy. Particularly problematic are *false positive* training instances, that arise when entity pairs for which a relation holds occur in contexts that do not express that relation. In this work we have mainly explored two approaches to this problem: (1) *at-least-one* learners, that assume that for each training entity pair at least one matching context actually expresses the relation, and (2) generative models that aim at separating the relational distribution from noise distributions. We have developed an *at-least-one* learner that incorporates a dedicated noise label and ranking constraints that make explicit negative training data unnecessary. Additionally, we extended an existing generative noise reduction model to incorporate feature-based representations. Both novel models as well as their combination improve upon non-noise-reduced base-lines and show state-of-the-art results.

The design of the overall Knowledge Base Bopulation system showed good benchmarking performance. Still, other approaches can also be imagined, for example relation prediction based on sequence labelers instead of classifiers. In initial experiments, we show promising results (better than the median performance in TAC KBP) using conditional random field taggers, however these models do not reach the top-performance achieved by the pipelined classification-based methods.

The entire code for *RelationFactory*, together with the trained models, is released as open source for facilitating further research. We hope to promote new ways of experimentation for researchers focusing on particular aspects of relation extraction.

As an outlook, evaluation of future relation extraction systems could also be improved if large-scale evaluation efforts, based on annotating pooled responses from benchmark participants, were designed to be better applicable to systems not included in the pool. We have suggested a more re-usable annotation scheme that would focus on only a small number of documents that contain the majority of answers, and which would the be extensively annotated for query-specific information.

To conclude, we aimed at advancing the state of the art in end-to-end Knowledge Base Population and characterized the complexity of the task both qualitatively and quantitatively. Since both well-performing algorithms and well-engineered systems are important foundations for future work in relation extraction, a focus was laid on both aspects. We hope that the growing number of use-cases for automatically structuring textual information can profit from this research, and that the work will be useful for future advances in relation extraction.

A. Appendix

A.1. Summary of TAC KBP Slot Descriptions

This is a list of relations with shortened relation definitions. The original relation definitions (slot descriptions) for TAC KBP 2012 (on which also TAC KBP 2013 is based on) comprise 30 pages and can be found at `http://nist.gov/tac/2012/KBP/task_guidelines/TAC_KBP_Slots_V2.4.pdf`.

- `per:alternate_names`: Names used to refer to the assigned person that are distinct from the "official" name.
- `per:date_of_birth`: The date on which the assigned person was born.
- `per:age`: A reported age of the assigned person.
- `per:country_of_birth`: The country in which the assigned person was born.
- `per:stateorprovince_of_birth`: The geopolitical entity at state or province level in which the assigned person was born.
- `per:city_of_birth`: The geopolitical entity at the municipality level (city, town, or village) in which the assigned person was born.
- `per:origin`: The nationality and/or ethnicity of the assigned person.
- `per:date_of_death`: The date of the assigned person's death.
- `per:country_of_death`: The country in which the assigned person died.
- `per:stateorprovince_of_death`: The geopolitical entity at state or province level in which the assigned person died.
- `per:city_of_death`: The geopolitical entity at the level of city, town, village in which the assigned person died.
- `per:cause_of_death`: The explicit cause of death for the assigned person.
- `per:countries_of_residence`: All countries in which the assigned person has lived.

- `per:statesorprovinces_of_residence`: Geopolitical entities at the state or province level in which the assigned person has lived.
- `per:cities_of_residence`: Geopolitical entities at the level of city, town, or village in which the assigned person has lived.
- `per:schools_attended`: Any school (college, high school, university, etc.) that the assigned person has attended.
- `per:title`: Official or unofficial name(s) of the employment or membership positions that have been held by the assigned person.
- `per:member_of`: The organization(s) of which the assigned person has been a member.
- `per:employee_of`: The organizations or geopolitical entities (governments) by which the assigned person has been employed.
- `per:religion`: The religion to which the assigned person has belonged.
- `per:spouse`: The spouse(s) of the assigned person.
- `per:children`: The children of the assigned person, including adopted and stepchildren.
- `per:parents`: The parents of the assigned person.
- `per:siblings`: The brothers and sisters of the assigned person.
- `per:other_family`: Family other than siblings, parents, children, and spouse (or former spouse): brothers-in-law, sisters-in-law, grandparents, grandchildren, cousins, aunts, uncles, etc.
- `per:charges`: The charges or crimes (alleged or convicted) of the assigned person.
- `org:alternate_names`: Any name used to refer to the assigned organization that is distinct from the "official" name, e.g. former names, aliases, alternate spellings, acronyms, abbreviations, translations, and any official designators such as stock ticker code.

- `org:political_religious_affiliation`: Ideological groups with which the organization is associated.
- `org:top_members_employees`: The persons in high-level, leading positions at the assigned organization. Top Member/Employee positions should imply a level of decision-making authority over the entire assigned organization.
- `org:number_of_employees_members`: The total number of people who are employed by or have membership in an organization.
- `org:members`: Organizations or Geopolitical entities that are members of the assigned organization. Correct fillers are distinct entities that are generally capable of autonomously ending their membership.
- `org:member_of`: Organizations or geopolitical entities of which the assigned organization is a member itself.
- `org:subsidiaries`: Organizations that are subsidiaries of the assigned organization. Subsidiaries are subsumed under the assigned organization, rather than being distinct entities.
- `org:parents`: Organizations or geopolitical entities of which the assigned organization is a subsidiary.
- `org:founded_by`: The person, organization, or geopolitical entity that founded the assigned organization.
- `org:date_founded`: The date on which the assigned organization was founded.
- `org:date_dissolved`: The date on which the assigned organization was dissolved.
- `org:country_of_headquarters`: Countries in which the headquarters of the assigned organization are located.
- `org:stateorprovince_of_headquarters`: Location of the headquarters of the assigned organization at the state or province level.

- `org:city_of_headquarters`: Location of the headquarters of the assigned organization at the city, town, or village level.
- `org:shareholders`: Any organization, person, or geopolitical entity that holds shares (majority or not) of the organization.
- `org:website`: An official top level URL for the organization's website.

A.2. List of Organization Suffixes

The base organizational suffixes are: *Co, Corp, Corporation, Inc, Incorporated, Industries, Limited, LLC, LLLP, LLP, LP, Ltd, Partners, PC, plc, Plc, PLC*

The system uses this list adding punctuation variants (e.g. an organization *"ORG"* is expanded to *"ORG LLLP"*, *"ORG, LLLP"*, *"ORG L.L.L.P."*, *"ORG, L.L.L.P."* etc.).

A.3. Per-relation Results

relation	precision	recall	F1-score
org:alternate_names	0.6571	0.2584	0.3709
org:city_of_headquarters	0.5294	0.3913	0.45
org:country_of_headquarters	1.0	0.1764	0.2999
org:date_founded	1.0	0.3076	0.4705
org:founded_by	0.666	0.0952	0.1667
org:number_of_employees_members	0.25	0.1818	0.2105
org:parents	0.6	0.2307	0.3333
org:stateorprovince_of_headquarters	0.5	0.05	0.0909
org:subsidiaries	0.3684	0.2121	0.2692
org:top_members_employees	0.6133	0.3538	0.4487
org:website	1.0	0.5625	0.72
per:age	0.5	0.3225	0.3921
per:alternate_names	0.8571	0.0967	0.1739
per:cause_of_death	0.9444	0.5151	0.6667
per:charges	0.6667	0.0444	0.0833
per:children	0.75	0.1052	0.1846
per:cities_of_residence	0.7777	0.132	0.2258
per:city_of_birth	1.0	0.4166	0.5882
per:city_of_death	0.5	0.125	0.2
per:countries_of_residence	1.0	0.0465	0.0888
per:country_of_birth	1.0	0.2	0.3333
per:date_of_birth	0.8571	0.4615	0.5999
per:date_of_death	0.0384	0.0277	0.0322
per:employee_or_member_of	0.4166	0.1219	0.1886
per:origin	1.0	0.125	0.2222
per:parents	0.6667	0.24	0.3529
per:schools_attended	0.4	0.0689	0.1176
per:siblings	0.5	0.25	0.3334
per:spouse	0.5336	0.2424	0.3333
per:title	0.4901	0.3363	0.3989

Table 37: Per-relation results, noise-reduced distant supervision patterns module. (All relations with at least one correct answer by the module are listed.)

relation	precision	recall	F1-score
org:city_of_headquarters	0.75	0.1304	0.2222
org:country_of_headquarters	1.0	0.0294	0.0571
org:number_of_employees_members	0.5	0.1818	0.2666
org:stateorprovince_of_headquarters	1.0	0.05	0.0952
org:subsidiaries	0.75	0.0909	0.1621
org:top_members_employees	0.5405	0.1538	0.2395
per:age	0.9	0.2903	0.439
per:alternate_names	1.0	0.0483	0.0923
per:cause_of_death	1.0	0.2121	0.35
per:charges	1.0	0.0222	0.0434
per:children	1.0	0.0526	0.0999
per:cities_of_residence	0.7777	0.1320	0.2258
per:city_of_birth	0.8333	0.4166	0.5555
per:city_of_death	0.6667	0.125	0.2105
per:countries_of_residence	1.0	0.0697	0.1304
per:country_of_birth	1.0	0.2	0.3333
per:country_of_death	1.0	0.2	0.3333
per:date_of_birth	0.8	0.3076	0.4444
per:date_of_death	0.0588	0.0277	0.0377
per:employee_or_member_of	1.0	0.0325	0.0629
per:origin	1.0	0.125	0.2222
per:parents	0.6	0.12	0.2
per:religion	1.0	0.1428	0.25
per:schools_attended	0.5	0.0344	0.0645
per:siblings	0.6	0.25	0.3529
per:spouse	0.6667	0.0606	0.1111
per:stateorprovince_of_birth	0.6667	0.2	0.3076
per:stateorprovince_of_death	1.0	0.0555	0.1052
per:statesorprovinces_of_residence	0.3333	0.0357	0.0645
per:title	0.4965	0.3228	0.3913

Table 38: Per-relation results, hand-written patterns module.

relation	precision	recall	F1-score
org:alternate_names	0.8333	0.337	0.48
per:alternate_names	0.6667	0.129	0.2162

Table 39: Per-relation results, alternate-names module.

relation	precision	recall	F1-score
org:alternate_names	1.0	0.0449	0.086
org:city_of_headquarters	0.5217	0.5217	0.5217
org:country_of_headquarters	0.4	0.3529	0.375
org:date_founded	0.8571	0.4615	0.5999
org:founded_by	0.625	0.2380	0.3448
org:member_of	0.5	0.25	0.3333
org:number_of_employees_members	0.2727	0.2727	0.2727
org:parents	0.375	0.4615	0.4137
org:stateorprovince_of_headquarters	0.625	0.25	0.3571
org:subsidiaries	0.5	0.0909	0.1538
org:top_members_employees	0.3945	0.4461	0.4187
org:website	1.0	0.5625	0.72
per:age	0.6875	0.3548	0.468
per:alternate_names	1.0	0.0483	0.0923
per:cause_of_death	0.862	0.7575	0.8064
per:charges	0.4285	0.0666	0.1153
per:children	0.4285	0.2105	0.2823
per:cities_of_residence	0.4	0.3773	0.3883
per:city_of_birth	0.4615	0.5	0.48
per:city_of_death	0.8636	0.5937	0.7037
per:countries_of_residence	0.5	0.0465	0.0851
per:country_of_death	1.0	0.2	0.3333
per:date_of_birth	0.75	0.2307	0.3529
per:date_of_death	0.04	0.0277	0.0327
per:employee_or_member_of	0.25	0.1138	0.1564
per:origin	0.5882	0.25	0.3508
per:other_family	1.0	0.0666	0.125
per:parents	0.5238	0.44	0.4782
per:religion	0.5	0.2857	0.3636
per:schools_attended	0.3809	0.2758	0.32
per:siblings	0.5454	0.5	0.5217
per:spouse	0.5555	0.303	0.3921
per:stateorprovince_of_death	0.5833	0.3888	0.4666
per:statesorprovinces_of_residence	0.5555	0.1785	0.2702
per:title	0.3609	0.547	0.4349

Table 40: Per-relation results, SVM classifier module.

relation	precision	recall	F1-score
org:alternate_names	0.7096	0.4943	0.5827
org:city_of_headquarters	0.5	0.5217	0.5106
org:country_of_headquarters	0.4333	0.3823	0.4062
org:date_founded	0.8571	0.4615	0.5999
org:founded_by	0.625	0.2380	0.3448
org:member_of	0.3333	0.25	0.2857
org:number_of_employees_members	0.2727	0.2727	0.2727
org:parents	0.375	0.4615	0.4137
org:stateorprovince_of_headquarters	0.625	0.25	0.3571
org:subsidiaries	0.3636	0.2424	0.2909
org:top_members_employees	0.3921	0.4615	0.4240
org:website	1.0	0.5625	0.72
per:age	0.5714	0.3870	0.4615
per:alternate_names	0.7368	0.2258	0.3456
per:cause_of_death	0.8965	0.7878	0.8387
per:charges	0.375	0.0666	0.1132
per:children	0.4285	0.2105	0.2823
per:cities_of_residence	0.4	0.3773	0.3883
per:city_of_birth	0.6153	0.6666	0.6400
per:city_of_death	0.84	0.6562	0.7368
per:countries_of_residence	0.7142	0.1162	0.2
per:country_of_death	1.0	0.3	0.4615
per:date_of_birth	0.8571	0.4615	0.5999
per:date_of_death	0.0370	0.0277	0.0317
per:employee_or_member_of	0.2714	0.1544	0.1968
per:origin	0.6315	0.3	0.4067
per:other_family	1.0	0.0666	0.125
per:parents	0.5238	0.44	0.4782
per:religion	0.75	0.4285	0.5454
per:schools_attended	0.3636	0.2758	0.3137
per:siblings	0.5454	0.5	0.5217
per:spouse	0.5	0.3333	0.4
per:stateorprovince_of_birth	0.6666	0.2	0.3076
per:stateorprovince_of_death	0.6666	0.4444	0.5333
per:statesorprovinces_of_residence	0.4545	0.1785	0.2564
per:title	0.3623	0.5605	0.4401

Table 41: Per-relation results, main run (merger of the classifier, manual patterns, noise-reduced patterns and alternate-names modules).

Bibliography

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE.

Alfonseca, E., Filippova, K., Delort, J.-Y., and Garrido, G. (2012). Pattern learning for relation extraction with a hierarchical topic model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 54–59. Association for Computational Linguistics.

Alfonseca, E., Pighin, D., and Garrido, G. (2013). Heady: News headline abstraction through event pattern clustering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1243–1253.

Angeli, G., Chaganty, A., Chang, A., Reschke, K., Tibshirani, J., Wu, J. Y., Bastani, O., Siilats, K., and Manning, C. D. (2013). Stanford's 2013 kbp system. In *Proc. Text Analysis Conference (TAC2013)*.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Balasubramanian, N., Soderland, S., Mausam, and Etzioni, O. (2012). Rel-grams: a probabilistic model of relations in text. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 101–105. Association for Computational Linguistics.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *International Joint Conference on Artificial Intelligence*.

Belanger, D., Passos, A., Riedel, S., and McCallum, A. (2014). Message passing for soft constraint dual decomposition. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*.

Belkin, M. and Niyogi, P. (2004). Semi-supervised learning on riemannian manifolds. *Machine Learning Journal*, 56(1-3):209–239.

Bentor, Y. (2013). University of texas at austin kbp 2013 slot filling system: Bayesian logic programs for textual inference. In *Text Analysis Conference (TAC 2013)*.

Blessing, A. and Schütze, H. (2012). Crosslingual distant supervision for extracting relations of different complexity. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1123–1132. ACM.

Bollegala, D., Kusumoto, M., Yoshida, Y., and Kawarabayashi, K.-I. (2013). Mining for analogous tuples from an entity-relation graph. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2064–2077. AAAI Press.

Bollegala, D., Matsuo, Y., and Ishizuka, M. (2011). Relation adaptation: learning to extract novel relations with minimum supervision. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 2205–2210. AAAI Press.

Bollegala, D. T., Matsuo, Y., and Ishizuka, M. (2010). Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *Proceedings of the 19th international conference on World wide web*, pages 151–160. ACM.

Borzsony, S., Kossmann, D., and Stocker, K. (2001). The skyline operator. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 421–430. IEEE.

Buckley, C. and Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32. ACM.

Buitelaar, P., Cimiano, P., and Magnini, B. (2005). *Ontology learning from text: methods, evaluation and applications*, volume 123. IOS press.

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

Bunescu, R. C. and Mooney, R. J. (2007). Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*.

Burchardt, A., Erk, K., Frank, A., Kowalski, A., Padó, S., and Pinkal, M. (2006). The salsa corpus: a german corpus resource for lexical semantics. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006)*.

Burchardt, A., Pennacchiotti, M., Thater, S., and Pinkal, M. (2009). Assessing the impact of frame semantics on textual entailment. *Natural Language Engineering*, 15(4):527–550.

Burchardt, A., Reiter, N., Thater, S., and Frank, A. (2007). A semantic approach to textual entailment: System evaluation and task analysis. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 10–15. Association for Computational Linguistics.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr, E. R., and Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, volume 5, page 3.

Cer, D. M., De Marneffe, M.-C., Jurafsky, D., and Manning, C. D. (2010). Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *Proceedings of the Conference on International Language Resources and Evaluation (LREC)*.

Chalupsky, H. (2012). Story-level inference and gap filling to improve machine reading. In *International Conference of the Florida Artificial Intelligence Research Society (FLAIRS)*.

Chalupsky, H. (2013). English slot filling with the knowledge resolver system. In *Text Analysis Conference (TAC 2013)*.

Chan, Y. S. and Roth, D. (2011). Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 551–560. Association for Computational Linguistics.

Chen, Z., Tamang, S., Lee, A., Li, X., Lin, W.-P., Snover, M., Artiles, J., Passantino, M., and Ji, H. (2010). Cunyblender tac-kbp2010 entity linking and slot filling system description. In *Proc. TAC 2010 Workshop*.

Chinchor, N., Lewis, D. D., and Hirschman, L. (1993). Evaluating message understanding systems: an analysis of the third message understanding conference (muc-3). *Computational linguistics*, 19(3):409–449.

Chrupała, G. and Klakow, D. (2010). A Named Entity Labeler for German: exploiting Wikipedia and distributional clusters. In *Proceedings of the Conference on International Language Resources and Evaluation (LREC)*, pages 552–556.

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), Volume 10*, pages 1–8. Association for Computational Linguistics.

Collins, M., Dasgupta, S., and Schapire, R. E. (2001). A generalization of principal components analysis to the exponential family. In *Advances in neural information processing systems (NIPS)*, pages 617–624.

Cormack, G. V., Palmer, C. R., and Clarke, C. L. (1998). Efficient construction of large test collections. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 282–289. ACM.

Craven, M., Kumlien, J., et al. (1999). Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, pages 77–86.

Doddington, G. R., Mitchell, A., Przybocki, M. A., Ramshaw, L. A., Strassel, S., and Weischedel, R. M. (2004). The automatic content extraction (ace) program-tasks, data, and evaluation. In *LREC*.

Downey, D., Etzioni, O., and Soderland, S. (2005). A probabilistic model of redundancy in information extraction. In *IJCAI*.

Downey, D., Schoenmackers, S., and Etzioni, O. (2007). Sparse information extraction: Unsupervised language models to the rescue. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 45, page 696.

Duh, K., Sudoh, K., Wu, X., Tsukada, H., and Nagata, M. (2012). Learning to translate with multiple objectives. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1–10. Association for Computational Linguistics.

Elson, D. K., Dames, N., and McKeown, K. R. (2010). Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147. Association for Computational Linguistics.

Etzioni, O., Banko, M., and Cafarella, M. J. (2006). Machine reading. In *AAAI*, volume 6, pages 1517–1519.

Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam, M. (2011). Open information extraction: The second generation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume One*, pages 3–10. AAAI Press.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.

Gancarz, M. (2003). *Linux and the Unix philosophy*. Digital Press.

Gardner, M., Talukdar, P. P., Kisiel, B., and Mitchell, T. (2013). Improving learning and inference in a large knowledge-base using latent syntactic cues. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*.

Giuliano, C., Lavelli, A., and Romano, L. (2006). Exploiting shallow linguistic information for relation extraction from biomedical literature. In *EACL*, volume 2006, pages 98–113.

Godfrey, P., Shipley, R., and Gryz, J. (2007). Algorithms and analyses for maximal vector computation. *The VLDB Journal—The International Journal on Very Large Data Bases*, 16(1):5–28.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235.

Grishman, R. (2013). Off to a cold start: New york university's 2013 knowledge base population systems. In *Text Analysis Conference (TAC 2013)*.

Haghighi, A. and Vanderwende, L. (2009). Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics.

Hamp, B. and Feldweg, H. (1997). GermaNet - a Lexical-Semantic Net for German. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15, Madrid, Spain.

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

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L. S., and Weld, D. S. (2011). Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, pages 541–550.

Hoffmann, R., Zhang, C., and Weld, D. S. (2010). Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 286–295. Association for Computational Linguistics.

Hovy, E. H. and Maier, E. (1995). Parsimonious or profligate: How many and which discourse structure relations. `http://www.isi.edu/natural-language/people/hovy/papers/93discproc.pdf`. Unpublished manuscript.

Illig, J., Roth, B., and Klakow, D. (2014). Unsupervised parsing for generating surface-based relation extraction patterns. In *EACL 2014*.

Ji, H. and Grishman, R. (2011). Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1148–1158.

Ji, H., Grishman, R., Dang, H. T., Griffitt, K., and Ellis, J. (2010). Overview of the tac 2010 knowledge base population track. In *Third Text Analysis Conference (TAC 2010)*.

Joachims, T. (1999). Making large scale svm learning practical.

Kasneci, G., Ramanath, M., Suchanek, F., and Weikum, G. (2009). The yago-naga approach to knowledge discovery. *ACM SIGMOD Record*, 37(4):41–47.

Kingsbury, P. and Palmer, M. (2002). From treebank to propbank. In *LREC*.

Kozareva, Z. and Hovy, E. (2010a). Learning arguments and supertypes of semantic relations using recursive patterns. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1482–1491. Association for Computational Linguistics.

Kozareva, Z. and Hovy, E. (2010b). Not all seeds are equal: Measuring the quality of text mining seeds. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 618–626. Association for Computational Linguistics.

Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119. ACM.

Landis, J. R., Koch, G. G., et al. (1977). The measurement of observer agreement for categorical data. *biometrics*, 33(1):159–174.

Lao, N. and Cohen, W. W. (2010). Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67.

Lao, N., Mitchell, T., and Cohen, W. W. (2011). Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics.

Lee, H., Recasens, M., Chang, A., Surdeanu, M., and Jurafsky, D. (2012). Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500. Association for Computational Linguistics.

Li, Y., Li, X., Huang, H., Song, Y., Chang, C., Zhou, L., Xiao, J., Yu, D., Xu, W., and Chen, G. (2011). Pris at tac2011 kbp track. In *Proceedings of TAC 2011 Workshop*.

Li, Y., Zhang, Y., Doyu Li, X. T., Wang, J., Zuo, N., Wang, Y., Xu, W., Chen, G., and Guo, J. (2013). Pris at knowledge base population 2013. In *Text Analysis Conference (TAC 2013)*.

Lin, D. and Pantel, P. (2001a). Dirt - discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM.

Lin, D. and Pantel, P. (2001b). Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

Lin, W., Yangarber, R., and Grishman, R. (2003). Bootstrapped learning of semantic classes from positive and negative examples. In *Proceedings of ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*, volume 1, page 21.

Louis, A. and Nenkova, A. (2012). A coherence model based on syntactic patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1157–1168. Association for Computational Linguistics.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

McDonald, R. T. and Nivre, J. (2007). Characterizing the errors of data-driven dependency parsing models. In *EMNLP-CoNLL*, pages 122–131.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990). Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3:235–244.

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

Min, B., Grishman, R., Wan, L., Wang, C., and Gondek, D. (2013). Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of NAACL-HLT*, pages 777–782.

Min, B., Li, X., Grishman, R., and Sun, A. (2012a). New york university 2012 system for kbp slot filling. In *Proceedings of the Text Analysis Conference (TAC)*.

Min, B., Shi, S., Grishman, R., and Lin, C.-Y. (2012b). Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1027–1037. Association for Computational Linguistics.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Mooney, R. J. and Bunescu, R. C. (2005). Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178.

Nakashole, N., Weikum, G., and Suchanek, F. (2012). Patty: a taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1135–1145. Association for Computational Linguistics.

Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9. ACM.

Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782.

Petukhova, V., Prévot, L., and Bunt, H. (2011). Multi-level discourse relations between dialogue units. In *Proceedings 6th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-6), Oxford*, pages 18–27.

Plank, B. and Moschitti, A. (2013). Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Poon, H. and Domingos, P. (2010). Unsupervised ontology induction from text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 296–305.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008). The penn discourse treebank 2.0. In *LREC*.

Ray, S. and Craven, M. (2005). Supervised versus multiple instance learning: An empirical comparison. In *Proceedings of the 22nd international conference on Machine learning*, pages 697–704. ACM.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press.

Reschke, K., Jankowiak, M., Surdeanu, M., Manning, C. D., and Jurafsky, D. (2014). Event extraction using distant supervision. In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC)*.

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013). Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL-HLT*, pages 74–84.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., and Gatford, M. (1995). Okapi at trec-3. *TREC-3 NIST Special Publication*.

Rocchio, J. J. (1971). *Relevance feedback in information retrieval*. Prentice-Hall, Englewood Cliffs NJ.

Rohanimanesh, K., Bellare, K., Culotta, A., McCallum, A., and Wick, M. L. (2011). Samplerank: Training factor graphs with atomic gradients. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 777–784.

Roth, B., Barth, T., Wiegand, M., Singh, M., and Klakow, D. (2013). Effective slot filling based on shallow distant supervision methods. In *Proceedings of the Sixth Text Analysis Conference (TAC 2013)*.

Roth, B., Chrupala, G., Wiegand, M., Singh, M., and Klakow, D. (2012). Generalizing from freebase and patterns using distant supervision for slot filling. In *Proceedings of the Text Analysis Conference (TAC)*.

Roth, B. and Klakow, D. (2010). Cross-language retrieval using link-based language models. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 773–774. ACM.

Roth, B. and Klakow, D. (2013). Feature-based models for improving the quality of noisy training data for relation extraction. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, CIKM '13, pages 1181–1184, New York, NY, USA. ACM.

Ruppenhofer, J., Sporleder, C., Morante, R., Baker, C., and Palmer, M. (2010). Semeval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 45–50. Association for Computational Linguistics.

Salton, G. and Buckley, C. (1997). Improving retrieval performance by relevance feedback. *Readings in information retrieval*, 24:5.

Sanderson, M. and Zobel, J. (2005). Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169. ACM.

Schlaefer, N., Ko, J., Betteridge, J., Pathak, M. A., Nyberg, E., and Sautter, G. (2007). Semantic extensions of the ephyra qa system for trec 2007. In *TREC*.

Segura-Bedmar, I., Martınez, P., and Herrero-Zazo, M. (2013). Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). *Proceedings of Semeval*, pages 341–350.

Singh, S., Yao, L., Belanger, D., Ari, K., Anzaroot, S., Wick, M., Passos, A., Pandya, H., Choi, J., Martin, B., and McCallum, A. (2013). Universal schema for slot filling and cold start: Umass iesl at tackbp 2013. In *Text Analysis Conference (TAC 2013)*.

Snow, R., Jurafsky, D., and Ng, A. Y. (2006). Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 801–808. Association for Computational Linguistics.

Soderland, S., Gilmer, J., Bart, R., Etzioni, O., and Weld, D. S. (2013). Open ie to kbp relations in 3 hours. In *Text Analysis Conference (TAC 2013)*.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Sun, A., Grishman, R., and Sekine, S. (2011). Semi-supervised relation extraction with large-scale word clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 521–529. Association for Computational Linguistics.

Surdeanu, M. (2013). Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *Text Analysis Conference (TAC 2013)*.

Surdeanu, M., Tibshirani, J., Nallapati, R., and Manning, C. D. (2012). Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics.

Takamatsu, S., Sato, I., and Nakagawa, H. (2012). Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 721–729, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tatu, M. and Moldovan, D. (2005). A semantic approach to recognizing textual entailment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 371–378. Association for Computational Linguistics.

U.S. Department of Agriculture (1992). Food guide pyramid. a guide to daily food choices. *Human Nutrition Information Service, Washington, DC, Home and Garden Bulletin*, (252).

Van Dongen, S. (2008). Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

Voorhees, E. M. (2000). Variations in relevance judgments and the measurement of retrieval effectiveness. *Information processing & management*, 36(5):697–716.

Voorhees, E. M. (2002). The philosophy of information retrieval evaluation. In *Evaluation of cross-language information retrieval systems*, pages 355–370. Springer.

Voorhees, E. M. and Buckley, C. (2002). The effect of topic set size on retrieval experiment error. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 316–323. ACM.

Wang, W., Su, J., and Tan, C. L. (2010). Kernel based discourse relation recognition with temporal ordering information. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 710–719. Association for Computational Linguistics.

Weischedel, R. and Brunstein, A. (2005). Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*.

Weston, J., Bordes, A., Yakhnenko, O., and Usunier, N. (2013). Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Joint Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Wiegand, M., Roth, B., and Klakow, D. (2012a). Web-based Relation Extraction for the Food Domain. In *Proceedings of the International Conference on Applications of Natural Language Processing to Information Systems (NLDB)*, pages 222–227, Groningen, the Netherlands. Springer.

Wiegand, M., Roth, B., Lasarcyk, E., Köser, S., and Klakow, D. (2012b). A Gold Standard for Relation Extraction in the Food Domain. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, pages 507–514, Istanbul, Turkey.

Xu, S., Zhang, C., Niu, Z., Mei, R., Chen, J., Zhang, J., and Fu, H. (2013a). Bit's slot-filling method for tac-kbp 2013. In *Proc. Text Analysis Conference (TAC2013)*.

Xu, W., Le Zhao, R. H., and Grishman, R. (2013b). Filling knowledge base gaps for distant supervision of relation extraction. In *ACL*.

Yao, L., Haghighi, A., Riedel, S., and McCallum, A. (2011). Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics.

Yao, L., Riedel, S., and McCallum, A. (2010). Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical*

Methods in Natural Language Processing, EMNLP '10, pages 1013–1023, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yu, D., Haibo, L., Cassidy, T., Li, Q., Huang, H., Chen, Z., Ji, H., Zhang, Y., and Roth, D. (2013). Rpi-blender tac-kbp2013 knowledge base population system. In *Text Analysis Conference (TAC 2013)*.

Yu, D., Huang, H., Cassidy, T., Ji, H., Wang, C., Zhi, S., Han, J., Voss, C., and Magdon-Ismail, M. (2014). The wisdom of minority: Unsupervised slot filling validation based on multi-dimensional truth-finding. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106.

Zhou, D., Huang, J., and Schölkopf, B. (2005). Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd international conference on Machine learning*, pages 1036–1043. ACM.

Zobel, J. (1998). How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314. ACM.