Akim Bassa

GerIE: Open Information Extraction for German Texts

Master's Thesis

Graz University of Technology

Knowledge Technologies Institute
Head: Univ.-Prof. Dr. Stefanie Lindstaedt

Advisor: Dipl.-Ing. Dr.techn. Mark Kröll
Assessor: Dipl.-Ing. Dr.techn. Roman Kern

Graz, July 2016

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, _____ _____

Date Signature

Eidesstattliche Erklärung[1]

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am _____ _____

Datum Unterschrift

[1] Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Abstract

Open Information Extraction (OIE) targets domain- and relation-independent discovery of relations in text, scalable to the Web. Although German is a major European language, no research has been conducted in German OIE yet. In this paper we fill this knowledge gap and present GerIE, the first German OIE system. As OIE has received increasing attention lately and various potent approaches have already been proposed, we surveyed to what extent these methods can be applied to German language and which additionally principles could be valuable in a new system. The most promising approach, hand-crafted rules working on dependency parsed sentences, was implemented in GerIE. We also created two German OIE evaluation datasets, which showed that GerIE achieves at least 0.88 precision and recall with correctly parsed sentences, while errors made by the used dependency parser can reduce precision to 0.54 and recall to 0.48.

Open Information Extraction (OIE) zielt auf domänen- und relationsunabhängige Erkennung von Relationen in Texten ab, skalierbar auf große Datensätze wie das Web. Obwohl Deutsch eine weitverbreitete europäische Sprache ist, gibt es bisher keine Arbeiten zu OIE für deutsche Texte. In dieser Arbeit füllen wir diese Wissenslücke und präsentieren GerIE, das erste deutsche OIE-System. Da OIE in letzter Zeit zunehmende Aufmerksamkeit erhalten hat und verschiedene leistungsfähige Ansätze vorgeschlagen wurden, haben wir diesbezüglich untersucht, inwieweit diese Methoden auch in der deutschen Sprache angewandt werden können und welche Prinzipien in einem neuen System sinnvoll wären. Der vielversprechendste Ansatz, manuell erstellte Regeln, die auf der Dependenz-Struktur von Sätzen basieren, wurde in GerIE umgesetzt. Für die Evaluierung haben wir zwei erste deutsche OIE Datensätze erstellt, deren Auswertungen zeigen, dass GerIE Precision- und Recallwerte von mindestens 0,88 bei korrekt geparsten

Sätzen erreicht, während vom Parser verursachte Fehler die Precision auf 0,54 und den Recall auf 0,48 reduzieren können.

Contents

List of Figures

1. Introduction

In traditional Information Extraction (IE) the desired relationships are always specified in advance, so the manual labour scales linearly with the number of specified relations. As this is not scalable to a large and heterogeneous corpus as the Web, Banko, M. J. Cafarella, et al. (2007) introduced the concept of *Open Information Extraction* (OIE), that aims to enable domain- and relation-independent discovery of relations. The idea of OIE is to learn how relations are expressed in general in a text, using unlexicalised features such as part-of-speech tags or dependency relations. These general patterns are however still language specific. OIE has various useful applications, such as question answering, opinion mining, fact checking or semantic full-text search. In recent years it has received increasing attention, and continuous research has improved performance of OIE systems constantly. Nearly all of the work so far has focused on English and although German is a major European language, no research has been conducted in German OIE yet. As OIE systems use language specific features, English systems are not applicable for German, also resources are, due to a smaller target audience, less available for German. For that reason we intend to fill this gap and develop a German OIE system. In the first step, we will survey existing methods and examine if and to what extend these methods can be applied to German language as well. This includes the investigation whether existing methods are expected to work with German grammar, whether similar performance can theoretically be expected and which (German) tools for preprocessing deliver best results. Additionally, we will examine what ideas for improvement have been suggested and consider their usefulness. We will use the acquired information for a prototypical implementation of a German OIE system. To evaluate the performance, we will determine what performance measures facilitate fair comparison, and create a first German OIE evaluation dataset. The published results and dataset should allow comparison with future systems.

The thesis is structured as follows: Chapter 2 provides an overview of Information Extraction and related work conducted in the field of Open Information Extraction. Chapter 3 ascertains an appropriate approach and architecture for a German OIE system and adequate resources necessary for preprocessing of German texts. Chapter 4 presents GerIE, a German Open Information Extraction system based on hand-crafted rules for dependency parsed sentences. Chapter 5 describes the evaluation of GerIE and discusses the results.

2. Background

This chapter provides an overview of Information Extraction, its evolution and the different tasks involved. Open Information Extraction and its application are described and the differences to traditional IE and other similar tasks pointed out. The reader is also given a digest of related work conducted in the field of OIE, especially for languages other than English.

2.1. Information Extraction

Piskorski and Yangarber (2013) defines information extraction as follows:

> "The task of Information Extraction is to identify instances of a particular pre-specified class of entities, relationships and events in natural language texts, and the extraction of the relevant properties (arguments) of the identified entities, relationships or events. The information to be extracted is pre-specified in user-defined structures called templates (or objects), each consisting of a number of slots (or attributes), which are to be instantiated by an IE system as it processes the text. The slots fills are usually: strings from the text, one of a number of pre-defined values, or a reference to a previously generated object template. One way of thinking about an IE system is in terms of database population, since an IE system creates a structured representation (e.g., database records) of selected information drawn from the analysed text."

2.1.1. Evolution of IE

Knowledge-Based Methods

The first information extraction systems emerged from the DARPA Message Understanding Conferences, where the participants were encouraged to develop systems to extract information from naturally occurring text. The systems of MUC-3 could be grouped into Pattern-Matching Systems, Syntax-Driven Systems and Semantics-Driven Systems (Chinchor, Lewis, and Hirschman, 1993). The top-performing systems did not spend time on automatic knowledge acquisition or learning techniques.

Supervised Methods

IE systems require extraction rules for each domain, so it was an important step to move away from Knowledge-Based Systems to systems which automatically learn an extractor from labelled training examples. Kim and Moldovan (1993), Riloff (1996), Craven et al. (2000) and Soderland (1999) use machine learning methods to extract domain-specific extraction patterns, which can be used to extract facts from text.

Weakly-Supervised Methods The creation of suitable training data still requires knowledge and time, so the next systems tried to reduce manual labour. Brin (1999), Riloff and Jones (1999), Agichtein and Gravano (2000) and Ravichandran and Hovy (2002) require for each relation only a small set of tagged seed instances or a few hand-crafted extraction patterns to begin the training process.

Self-Supervised Methods A self-supervised system does not need hand-tagged training data; instead, it learns to label its own training examples using a small set of domain-independent extraction patterns. Self-supervised systems are a species of unsupervised systems. Etzioni, M. Cafarella, et al. (2005) developed the *KnowItAll Web IE system*, which was self-supervised and domain-independent, but still needed a set of relations listed by the user

beforehand. Rosenfeld and Feldman (2006) used the same approach, with the only exception that they generated their extraction patterns with the input (description of target relations) and a collection of web pages. Daniel S. Weld et al. (2008) showed how to use Wikipedia and its infoboxes to automatically train an extractor. They matched sentences with the corresponding attributes in the infobox to automatically create a training dataset, but this approach restricts the target relations to those existing the infoboxes, so it is not ready for application on the Web.

2.1.2. Task Types

The term IE describes the process of extracting structured information from unstructured or semi-structured text. This includes several tasks, such as Named Entity Recognition, Co-reference Resolution or Event Extraction.

- **Named Entity Recognition (NER)** aims to identify proper names in free text, and to classify those entities into a set of predefined categories. Common categories are persons, organisations and locations, but also smaller groups such as expressions of times or measures (monetary values, percentages...) are sometimes involved. NER is not an easy task because named entities may be difficult to find and to categorise. For example the same name can be used to describe an Organisation and a Location ("France won the European championship." vs. "The European championship took place in France.").
- **Co-reference Resolution** involves the connection of various references in a text to the same entity. Pronouns, for example, have to be connected to the referred entity in order for the text to be interpreted correctly. In the sentence "Bill read his book." "his" may refer to "Bill" or another male mentioned previously, which shows the complexity of this problem.
- **Relation Extraction (RE)** addresses the detection and classification of relationships between entities, typically from machine readable text. The relationships here are predefined, for example *PresidentOf(subject,organisation)*.
- **Event Extraction** is the task of extracting information concerning incidents which are referred to in the text. Usually the questions *who*,

what, *where*, *when*, *why*, *how* are targeted. This is useful in order to, for example, receive structured information about terrorist actions from news, which was a task in MUC-3 and MUC-4[1].

2.2. Open Information Extraction

Open Information Extraction was introduced in 2006 by Banko, M. J. Cafarella, et al. (2007), to tackle the challenge of Web extraction. The Web has several properties which make traditional IE ineligible for this task. Firstly, it contains all possible kinds of domains and article types, whereas most IE work has concentrated on specific domains. Secondly, the relations of interest in the Web are often unknown and the number is high, which also makes the use of IE with its predefined relations impractical. Lastly, the Web contains billions of documents, which means that a system would have to apply highly scalable extraction techniques. Thus, Banko, M. J. Cafarella, et al. (2007) described three properties, which were considered mandatory when extracting information from the web corpus: *domain independence*, *automation* and *efficiency*. Adherence to these properties should ensure that the challenges which arise when trying to extract information from a massive and heterogeneous corpus can be handled. They successfully proved this with the TextRunner System (Banko, M. J. Cafarella, et al., 2007). All following Systems (ArgOE (Gamallo and Garcia, 2015), CSD-IE (Bast and Haussmann, 2013), ClausIE (Del Corro and Gemulla, 2013), OLLIE (Schmitz et al., 2012) to name a few) acknowledged these specifications.

Independence Domain independence is a property which differentiates OIE from traditional IE, which focused on specific fields. The fact that the web contains all possible kinds of genres and topics makes it essential that an OIE System is independent of the domain.

[1] http://www-nlpir.nist.gov/related_projects/muc/

Automation In Traditional IE Systems, the relations which should get extracted must be known and specified beforehand, and for each of these relations, manual effort like hand-crafted extraction patterns or hand-labelled training examples is required. The problem with a huge corpus like the web is, that the relations of interest are unanticipated, we do not know how many and which relations exist. OIE does not aim to extract specific relations, but as many relations as possible. This can be achieved with the help of a model which describes how relationships are expressed in general.

Efficiency Due to the vast and ever-growing number of web-pages efficiency is an important property of each OIE system. Here the ability of an OIE system to just extract all relations without the need to know them is very important. As a consequence, there is no need to repeat the extraction process for a newly defined relation, like in traditional IE systems.

2.2.1. Related Work

A variety of systems and approaches have been proposed since OIE was introduced, the majority of them designed for the English language. In addition Chinese, Spanish and Romance languages have already been addressed, but German is yet to be researched in terms of OIE.

English Banko, M. J. Cafarella, et al. (2007) proposed the TextRunner system, which used a Naive Bayes classifier to train a model based on shallow features and could then extract triples in a single pass over a corpus. Wanderlust (Akbik and Broß, 2009) was the first to utilise deep syntactic parsing in the form of link grammar (Sleator and Temperley, 1995), it automatically learned 46 patterns from an annotated corpus of 10,000 sentences. Wu and Daniel S Weld (2010) described their systems WOE^{pos}, working with shallow features to train Conditional Random Fields (CRF), and WOE^{parse}, which used features from dependency-parse trees and a pattern learner to decide whether the shortest path between two noun phrases expresses a relation. Unlike TextRunner, they have a high-quality training corpus obtained from Wikipedia (by automatically matching the infobox attribute values to

corresponding sentences). With their direct comparison of WOE^{parse} and WOE^{pos}, they showed that dependency parse features increase precision and recall compared to shallow features. StatSnowball (Zhu et al., 2009) also uses shallow parsing techniques, as they are cheaper and more robust, but it sees pattern selection as a problem of structure learning in Markov logic networks (Kok and Domingos, 2005). Fader, Soderland, and Etzioni (2011) proposed ReVerb, the successor of TextRunner, which aims to prevent frequent errors from TextRunner, incoherent and uninformative extractions. For this, they articulated syntactic and lexical constraints on binary, verb-based relation phrases, which yielded more informative relations. Christensen, Soderland, Etzioni, et al. (2010) observed that semantically labelled arguments in a sentence very often match the arguments in OIE extractions, and the verbs often correspond to the OIE relations. Thus, they proposed a system which converts the output of a semantic role labelling (SRL) system to OIE facts. This approach showed to yield lower precision for highly redundant text (as it is in the Web), while being over 2 orders of magnitude slower compared to TextRunner, so it was not further researched. Kraken (Akbik and Löser, 2012) was build upon their previous work Wanderlust, which showed that a limited number of patterns is sufficient for deep syntactic parsed sentences. That is why Kraken uses dependency parsing with hand-crafted rules. Schmitz et al. (2012) accept the trend of using dependency-parse features and presented OLLIE, the successor of ReVerb. OLLIE uses high precision tuples from ReVerb to bootstrap a training set for their pattern learner. In contrast to previous OIE systems, it also extracts relations mediated by nouns or adjectives, and includes essential contextual information in the extractions (for example when the relation is within a belief or conditional context). Nakashole, Weikum, and Suchanek (2012) applied OIE with the intent to organise the extracted relations into synsets and a taxonomy in WordNet-style. They apply dependency parsing and named entity recognition to extract a relation and assign a pattern synset, such as *<Politician>politician from <State>*. Del Corro and Gemulla (2013) introduce the clause-based approach implemented in ClausIE, where the detection and generation of facts is separated. They also work with hand-crafted rules utilizing the dependency structure of a sentence. Further, they identify the type of clauses according to the grammatical function of its constituents. They exploit this knowledge to generate multiple propositions out of a single clause. LSOE (Castella Xavier et al., 2013) was the first system which aimed

to use hand-crafted rules for POS-tagged texts. They utilise Qualia structure (Cimiano and Wenderoth, 2005), which provides information about the role of words in a sentence. Bast and Haussmann (2013) applied a technique called contextual sentence decomposition to decompose a sentence into pieces which semantically belong together. They employ rules to convert the output of a constituent parser to a Sentence-Constituent-Identification tree. They showed that the new representation allows easy extraction of various types of relations. Due to the reason that most of the OIE systems focused only on verb-mediated relations, Xavier and Lima (2014) proposed a method to enrich a text in such a way so that common OIE systems will also extract noun compounds ("glass vase") and adjective-noun pairs ("raw food"). The suggested idea here was to replace the phrase with a phrase which contains a verb and has the same meaning. ReNoun (Yahya et al., 2014) also just focuses on the extraction of noun-mediated relations, because of the lack of work done in this area.

Other Languages Gamallo, Garcia, and Fernández-Lanza (2012) showed that OIE based on dependency trees is suitable for various languages. They used a multilingual parser with a common output tagset for the supported languages (English and Romance languages). The improved multilingual OIE system ArgOE (Gamallo and Garcia, 2015) tried to be more open for different dependency parsers by using the CoNLL-X format. Due to the mediocre performance of the multilingual parsers, their results were not as good as those from other dependency based OIE systems. Zhila and Gelbukh (2013) described the Spanish system ExtrHech, working with POS-tagged input and semantic constraints, demonstrating that this approach achieves similar results in Spanish and in English. Wang, Li, and Huang (2014) applied OIE on Chinese articles, but decided to use a semi-supervised approach and focused on a fixed set of entities, namely person, organisation, location and time.

2.2.2. Application

A variety of applications for OIE exist, which shows the importance of developing a German OIE system. First of all **Question Answering** comes

to mind, because the triple representation allows to easily search for one or two missing components when the facts are stored in a relational database, for example. The question "Who is the president of America?" may translate to [??][president of][America]. Such a system was already implemented using extractions from ReVerb from over a billion web pages, available at http://openie.allenai.org/. Another application could be intelligent indexing for search engines. OIE provides information about the content of web pages independently of the domain, which can be used to intelligently index those pages. Bast, Bäurle, et al. (2012) presented a **semantic full-text search** engine called Broccoli, which adds the benefits of ontology search to full-text search. A query like *"list of presidents wearing glasses"* is hard for a normal full-text search, but easy when OIE was applied. This application is similar to Question Answering, but Question Answering is often required to answer questions in natural language. The facts extracted by OIE can also be used for **Sentiment Analysis**. Sentiment Analysis is the task of collecting and categorizing opinions about a topic or a product. OIE would help here with the collection of information, the categorisation (such as negative/positive statement) has to be done separately. Sentiment Analysis is especially attractive for companies because it obviates the need for conducting a survey. **Fact Checking** (the process of deciding whether to believe a fact or not) also profits from OIE, because it provides a huge amounts of facts from all kinds of sources, which is essential for many fact checking algorithms. The most basic approach is to take a vote, a claim backed by many different sources may be seen as true (Pasternack and Roth, 2010).

2.2.3. Open Versus Traditional IE

Banko, Etzioni, and Center (2008) considered this question and compared their OIE system O-CRF to a traditional IE system. They found that O-CRF achieves high precision and recall for a set of 500 sentences without requiring manually labelled training data for each relation. The traditional system, however, required hundreds to thousands labelled examples in three of four cases to achieve similar precision. Their conclusion was that OIE is essential when the relationships in a corpus are unknown or the number is huge, and

even for a small set of target relations traditional IE is only required when high recall is desirable. Soderland et al. (2010) addressed another problem of OIE. The paradigm that it is domain and relation independent leads to purely textual extractions, which are inadequate for ontologies. Thus, they described an approach to adapt OIE to a domain-specific ontology, which required a use of 10 training examples for each domain relation. Nakashole, Weikum, and Suchanek (2012) showed with the system PATTY that existing knowledge bases can be harnessed for entity-type information, so that patterns can be organised into synsets and a taxonomy.

2.2.4. Similar Tasks

Lifelong Knowledge Extraction This concept describes a system which automatically extracts information from the Web. The extraction process is continuous, the system runs forever, and it should learn from the gained knowledge to improve its extraction process. ALICE (Banko and Etzioni, 2007) and NELL (Carlson et al., 2010) are both such systems. The difference from OIE is that Lifelong Knowledge Extraction concentrates not only on relations, but also on various kinds of knowledge (like concepts from WordNet). Thus, it can employ various extraction techniques, including OIE. The fact that it learns more patterns every day forces such a system to repeatedly process the same text because it might contain previously undetected information. Naturally it also focuses on high precision, since recall should increase automatically over time.

Semantic Role Labelling SRL has the goal to identify the semantic arguments associated with a verb in the sentence. It also tries to classify those arguments into specific roles, such as Agent, Patient or Instrument. SRL begins with the verb and then locates its arguments, while OIE can have different approaches (for example, it searches for a phrase which constitutes a relation between two entities). The relation can also be more than a single verb ("is the brother of"). OIE and SRL are still similar, Christensen, Soderland, Etzioni, et al. (2010) showed how to convert the output of a SRL system to OIE facts.

3. Open IE Analysis

The purpose of this chapter is to ascertain an appropriate approach and architecture for a German OIE system and adequate resources necessary for preprocessing of German texts. GerIE, our prototype described in chapter 4, is based on the results obtained here. To accomplish this, we survey existing approaches for OIE, analyse their performance and applicability to German texts. We compare available dependency parsers (required for preprocessing) and finally investigate how evaluation should be conducted.

3.1. Existing Open IE Systems

The core component of all existing OIE systems is an *Extractor*. This component gets somehow preprocessed sentences as input, applies patterns to this input to extract facts, and provides these facts as output. Table 3.1 shows which techniques existing systems use.

3.1.1. Input

All of the OIE systems either operate with POS tags and other shallow features, or expect sentences with dependencies between the words, which requires more complex deep linguistic analysis. Certain patterns which express relations between entities exist for both sentences with POS tags or dependency labels.

Table 3.1.: Input and approach of existing OIE systems.

System	Input	Pattern Creation	Trained
TextRunner (Banko, M. J. Cafarella, et al., 2007)	PoS, NP-chunks	Naive Bayes classifier	✓
Wanderlust (Akbik and Broß, 2009)	link grammar	pattern learner	✓
WOEparse (Wu and Daniel S Weld, 2010)	dependencies	pattern learner	✓
WOEpos (Wu and Daniel S Weld, 2010)	PoS, NP-chunks	CRF	✓
StatSnowball (Zhu et al., 2009)	PoS	Markov logic networks	✓
ReVerb (Fader, Soderland, and Etzioni, 2011)	PoS, NP-chunks	syntactic and lexical constraints + logistic regression classifier	✓
SRL-IE (Christensen, Soderland, Etzioni, et al., 2010)	SRL	rule-based conversion	
DepOE (Gamallo, Garcia, and Fernández-Lanza, 2012)	dependencies	hand-crafted rules	
Kraken (Akbik and Löser, 2012)	dependencies	hand-crafted rules	
OLLIE (Schmitz et al., 2012)	dependencies	Open Pattern Learning	
Patty (Nakashole, Weikum, and Suchanek, 2012)	dependencies	frequent itemset mining	✓
ClausIE (Del Corro and Gemulla, 2013)	dependencies, constituents	hand-crafted rules	
LSOE (Castella Xavier et al., 2013)	PoS	Qualia Structure Based Patterns(Cimiano and Wenderoth, 2005)	
CSD-IE (Bast and Haussmann, 2013)	constituents	hand-crafted rules	
ArgOE (Gamallo and Garcia, 2015)	dependencies	hand-crafted rules	
ReNoun (Yahya et al., 2014)	dependencies, NP-chunks, NER	pattern learner	
SCOERE (Wang, Li, and Huang, 2014)	dependencies, constituents, NER	CRF	✓
BoostingOIE (Xavier and Lima, 2014)	PoS, NP-chunks	hand-crafted rules	
TK (Y. Xu et al., 2013)	dependencies	SVM tree kernels	✓
ExtrHech (Zhila and Gelbukh, 2013)	PoS	hand-crafted rules	

Part-of-speech Tags

A POS tagger categorises items of a sentence (words, punctuation marks, numerals...), so that each item has a certain POS tag assigned. A tag represents a certain category of items which have related grammatical properties, such as verb, noun, pronoun, adjective...

Banko (2009) stated multiple reasons why POS tagger should be favoured over dependency parsers in OIE: Firstly, the accuracy of POS taggers is better than those of parsers. English state-of-the-art taggers achieve an accuracy over 97% (Horsmann, Erbs, and Zesch, 2015), whereas English dependency parsers are more domain dependent, which leads to a varying accuracy, sometimes over 90% and sometimes down to 50% (Choi, J. Tetreault, and Stent, 2015). Another reason is the use of redundancy-based methods (Brill, 2003), which can improve the result of OIE systems relying on POS taggers. The idea is that the web corpus is highly redundant, and the same piece of information will be present in differently phrased statements. This increases the chance that one of the phrases has a POS pattern which can be extracted. The major advantage of sticking to shallow features like POS is the performance. Since one of the paradigms is efficiency, execution time is seen as important, and POS taggers are a magnitude faster than dependency parsers. WOE(Wu and Daniel S Weld, 2010), who implemented both approaches, reported that the POS approach was about 30 times faster).

Noun Phrase Chunking

A noun phrase chunker splits sentences into individual noun phrases that do not contain other noun phrases.

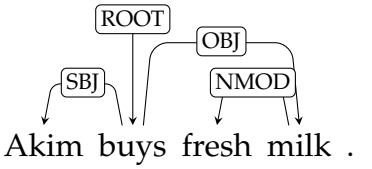
Barack Obama is a capable president.
[Barack Obama] is [a capable president].

Noun phrases belong to shallow features and are critical information when working with POS tags. They simplify the sentence structure and help to identify entities. The systems which restrict themselves to shallow features either use POS + NP-chunks (TextRunner, WOEpos, ReVerb...) or only POS and detect noun phrases themselves (LSOE, ExtrHech). ExtrHech, for

example, has four regular expressions to find noun phrases with help of the POS tags. State-of-the-art chunker achieve precision and recall over 90% [1].

Dependency Relations

Dependency relations are directed links between tokens (words, punctuation marks) in the sentence. Each token is connected to exactly one other token, and the label of the link between those tokens describes the grammatical relation (Subject, Predicate, Relative Clause...). An additional rule for a well-formed dependency structure is that exactly one root exists, usually the main verb of the sentence. This leads to a tree-structure which represents the sentence. There is a distinction between projective (edges may not cross) and non-projective (edges may cross) dependency trees.

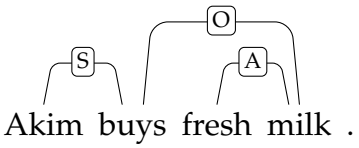


Although Banko, M. J. Cafarella, et al. (2007) applied dependency parsing to automatically create a proper training dataset, Wanderlust(Akbik and Broß, 2009) was the first OIE system which used deep linguistic analysis for the extraction task, and not only training. They parsed input sentences to get the Link Grammar (see 3.1.1), which is similar to the dependency grammar. They believed that the high costs in terms of time and resource consumption will be of minor importance, due to "cheap and easily accessible compute clusters". After WOE (Wu and Daniel S Weld, 2010) showed in a direct comparison that dependency features enable significantly higher precision and recall, dependency relations received more attention and were utilised in systems like OLLIE, Patty, ClausIE, ArgOE and ReNoun.

[1] http://aclweb.org/aclwiki/index.php?title=NP_Chunking_(State_of_the_art)

Link Grammar

A Link Grammar (Sleator and Temperley, 1995) describes relations between pairs of words. In contrast to a dependency structure, the links are undirected, may form circles and there is no root node.



Constituency Relations

A constituent parser divides a sentence into constituents. A constituent tree consists of terminals and non terminals, the tokens of the sentence are the leaf nodes and can be labelled with terminal categories (PoS tags). The interior nodes describe the constituency relations. This is very similar to chunking, the difference is that constituent parser attempts to find all constituents and go deeper into the sentence (hence this is categorised as deep syntactic analysis), whereas NP-chunker only tries to find the base noun phrases. Because of this, NP-chunks do not require a head, while constituents always have a head:

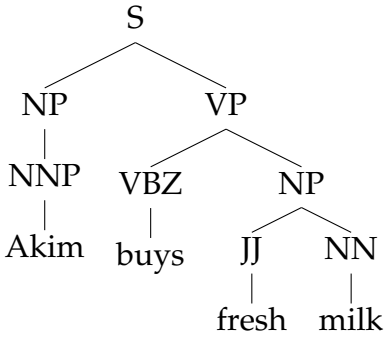


Table 3.2.: NomBank Example

```
[Arg0(agent) They] complained [Arg1(topic) about that issue].
```

Semantic Role Labels

Semantic role labelling tries to identify the semantic arguments associated with a verb in the sentence, and to classify those arguments into specific roles, such as Agent, Patient, Instrument, etc.

Christensen, Soderland, Etzioni, et al. (2011) observed that "verbs and their semantically labelled arguments almost always correspond to Open IE relations and arguments respectively." For example, the agent in the example provided in table 3.2 corresponds to the subject, and the topic corresponds to the object. Thus, Christensen, Soderland, Etzioni, et al. (2011) used rules to convert labelled SRL extractions to OIE extractions.

3.1.2. Pattern Creation

All extractors exploit general patterns in the grammatical structure of a sentence to extract relations. These patterns are either hand-crafted or the extractor is trained with labelled data.

Training Data

Eleven out of twenty OIE systems listed in table 3.1 used annotated data to train the extractor. The decision of training the extractor has two reasons: Firstly, it reduces the amount of manual labour (because there is no need to craft the extraction rules by hand), given that the training data do not have to be labelled manually. Secondly, an unforeseen number of possible relations in the web corpus may lead to a huge number of patterns, which cannot be captured all in advance. To minimise required manual work, different methods were applied to automatically create a training dataset.

- **Dependency parsing** TextRunner used dependency parsing to label its own training data. While only using shallow features when extracting facts at web scale, Banko, M. J. Cafarella, et al. (2007) hypothesised that "a parser can help to train an Extractor."
- **Wikipedia** WOE relies on Wikipedia and the structured data provided in the infoboxes to automatically create training examples. WOE's *matcher* heuristically matches attribute-value pairs of the infoboxes of an article with the corresponding sentences.
- **NER** Patty applies Named Entity Recognition to the training text, and uses the shortest dependency path between two entities in a sentence as example relation.
- **Seed facts** ReNoun first extracts a small number of seed facts using high-precision extractors. These seed facts are utilised to learn dependency parse patterns with distant supervision (Mintz et al., 2009). Other extractors such as StatSnowball or OLLIE which are successors of a previous system also used high precision seed tuples of their predecessors.

Although these methods reduce required efforts, they may introduce some flawed training samples, which negatively affects the training of the extractor. Annotating the data manually ensures good quality of the training set, while posing much work. Scoere and Wanderlust chose this approach, respectively with 539 and 4005 manually annotated sentences.

Technique
Several techniques have been applied to train a model for the extractor:

- **Naive Bayes** TextRunner utilised Naive Bayes for their extractor, which did not establish in OIE systems, because other techniques yielded better outcome.
- **Conditional Random Fields (CRF)** Here, the extraction problem is treated as a sequence-labelling task, which was shown to work well (Wu and Daniel S Weld, 2010; Etzioni, Fader, et al., 2011; Wang, Li, and Huang, 2014) and got better results compared to the Naive Bayes model.
- **Markov logic networks** StatSnowball implemented general discriminative Markov logic networks (Richardson and Domingos, 2006), a combination of logistic regression (LR) and CRF.

And for depencency parsing:

- **(Open) pattern learning** OLLIE and ReNoun bootstrap data based on seed tuples to automatically learn relation-independent dependency parse-tree pattern.
- **Frequent itemset mining** Nakashole, Weikum, and Suchanek (2012) applied the technique described by Srikant and Agrawal (1996). They viewed sentences as "shopping transactions", and each transaction has a "purchase" of several triples. The triple combinations with high co-occurrence support are computed by the mining algorithm.
- **Tree kernels** Y. Xu et al. (2013) adapted an SVM dependency tree kernel model (Moschitti, 2006), which achieved superior results compared to OLLIE and ReVerb.

Hand-crafted

Nine of the surveyed OIE systems relied on hand-crafted rules to extract facts from natural text. The groundwork was done by Akbik and Broß (2009), the authors annotated relation triples in 4005 sentences, so they could automatically obtain the link paths of these relations. All of the extracted paths were then applied again on the training set, so they could count the number of true positives and false positives for each linkpath. This measure was used at the level of confidence for facts extracted by the corresponding linkpath. A total of 46 valid linkpaths were found. The insights received with Wanderlust were used in the successor Kraken to manually create extraction rules based on dependency parse information. The state-of-the-art systems ClausIE, CSD-IE, ArgOE and ReNoun continued the trend of using a hand-crafted rule set for dependency parsed sentences.

In contrast to dependency parsed sentences where the number of rules is small, manual rule creation for POS tagged sentences is more difficult. Castella Xavier et al. (2013) were the first to claim that it is not necessary to use a large list of patterns, and showed with their system LSOE that a few hand-crafted lexical-syntactic patterns achieve similar results. They made use of the Qualia structure (Pustejovsky, 1991), which specifies four aspects of an object: the Constitutive Role (the relation between it and its constituent parts), Formal Role (that distinguishes it within a larger domain), Telic Role

(its purpose and function), and its Agentive Role (whatever brings it about). LSOE automatically identifies these roles in POS tagged text with patterns described by Cimiano and Wenderoth (2005).
Xavier and Lima (2014) described hand-crafted rules for a very specific area, namely to learn relations as the ones within noun compounds (glass vase) and adjective noun pairs (raw food), which were not addressed in previous systems.
ExtrHech (Zhila and Gelbukh, 2013) extracted relations by applying syntactic and lexical constraints on POS tagged input. The described expression for a verb phrase was for example $VREL \rightarrow (VW * P)|(V)$.

3.1.3. Output

A fact usually consists of a subject, a predicate which represents the relation, and an arbitrary number of objects:

[Albert Einstein][died][]
[Albert Einstein][died][in 1955]
[Albert Einstein][died][[in 1955][in Princeton]]

Nearly all examined OIE systems stored the facts as **triples**. To do this, the number of objects has to be reduced to one, which can be done, for example, by merging multiple object into one:

[Albert Einstein][died][in 1955 in Princeton]

or generating a distinct fact for each object:

[Albert Einstein][died][in 1955]
[Albert Einstein][died][in Princeton]

or both. Kraken(Akbik and Löser, 2012) decided to keep the natural **n-ary** structure of facts, because the reduction may lead to crucial information loss: In *[Albert Einstein][moved to][[Munich][in 1880]]* the separation of the two objects may lead to the loss of information *when* he moved to Munich.

Another aspect is the information included additionally to the surface form. Gamallo, Garcia, and Fernández-Lanza (2012) pointed out that it is important to provide additional information which was obtained from the

dependency parser, for example. The given reason was that "substantial postprocessing is needed to derive relevant linguistic information from the tuples".

3.2. Additional Concepts

In addition to the original paradigms, further principles were applied in newer systems. These are not essential for OIE, but may prove useful when developing a new system.

3.2.1. Minimality

CSD-IE(Bast and Haussmann, 2013) aims to extract facts to be minimal. This means that a fact should not contain other facts. In the sentence *"President Barack Obama was born in the USA."* two minimal extractions would be *[Barack Obama][is][President]* and *[Barack Obama][was born][in the USA]*. A non-minimal fact would be *[President Barack Obama][was born][in the USA]*, which should be avoided. Minimality is also necessary when the information of a separated fact cannot be excluded from another fact. In this case, the excluded fact may be referenced:

#1: *[Obama][said][that #2]*
#2: *[America][is not][a Christian nation]*

Two reasons why minimality should be incorporated were mentioned: the use of extracted facts in semantic full-text search and easier transformation of OIE triples into disambiguated facts within a formal ontology. No other OIE System mentions minimality explicitly.

3.2.2. Levels of Granularity

Levels of granularity describe how the extracted fact is stored or provided for further use. Gamallo, Garcia, and Fernández-Lanza (2012) emphasise that "substantial postprocessing is needed to derive relevant linguistic

information from the tuples" which is why it is not enough to output triples in textual form. Therefore, DepOE (Gamallo, Garcia, and Fernández-Lanza, 2012) additionally provides syntax-based information, POS tags, lemmas and heads. The successor of DepOE, ArgOE (Gamallo and Garcia, 2015), also keeps this property.

3.2.3. Separation of Detection and Representation

This property affects the architecture of OIE systems, as it allows customised generation of propositions from the detected clauses. This approach was first presented by Del Corro and Gemulla (2013) and implemented in their system ClausIE. In the detected clause *"Barack Obama was born in the USA."* following propositions may be generated:

[Barack Obama][was born][]
[Barack Obama][was][born]
[Barack Obama][was][born in the USA]
[Barack Obama][was born][in the USA]
[Barack Obama][was born in][the USA]

It is not necessary to stick with triples to represent facts, the representation can easily be changed without affecting the detection. CSD-IE uses a similar approach: it also first decomposes sentences into their basic constituents and afterwards create triples from those constituents.

3.2.4. Separation of Relation Detection and Relation Extraction

Y. Xu et al. (2013) addressed the task of determining whether there is a relation between a pair of entities in the sentence or not, before trying to extract the information. The authors pointed out that previous OIE systems ignore this question and report conflicting results for their systems. This task is difficult because it is not always clear what a relation constitutes. For instance in the phrase *"Obama eats apple pie."*: is there a relation between *Obama* and *apple*?

3.2.5. Context Analysis

OLLIE (Schmitz et al., 2012) introduced the new processing step context analysis, which adds additional context information like attribution and clausal modifiers. This means that the traditional *subject/predicate/object* triple will be extend with a new field which contains information about the context:

*[Barack Obama][was not born][in the USA][**AttributedTo claims; Donald Trump**]*

OLLIE finds this context information with help of the dependency parse structure, a list of communication and cognition verbs from VerbNet (Schuler, 2005) and lexical features. Previous OIE systems did not consider the context when extracting facts, which led to incorrect extractions:

[Barack Obama][was not born][in the USA]

3.2.6. Confidence Score

A confidence score is a measure assigned to each extracted fact, which states how confident the system is that the fact is correct. It is a way to trade recall for precision, by setting a confidence threshold.

This score was already provided in the first OIE system, TextRunner (Banko, M. J. Cafarella, et al., 2007), with assistance of a simple algorithm. The redundancy-based assessor in TextRunner just counts the number of distinct sentences in which a certain extraction occurs, and uses this count as a measure for the correctness of the extraction.

ReVerb (Fader, Soderland, and Etzioni, 2011) calculates the confidence score with a logistic regression classifier, using 19 features (for example: *(x, r, y)* covers all words in *s*; The last preposition in *r* is *for*).

OLLIE (Schmitz et al., 2012) made use of the same classifier as ReVerb, but with different features, now including information about the *AttributedTo* or *ClausalModifier* fields (described in 3.2.5).

WOE (Wu and Daniel S Weld, 2010) does not compute an additional confidence score after the extractions are made, but its pattern classifier calculates the normalised logarithmic frequency of the pattern of a triple, and this value is then used as confidence score.

Yahya et al. (2014) describes a different approach which was implemented in ReNoun. First, it assigns a score to a pattern, depending on the semantic similarity of the attributes of its extracted facts (a high similarity leads to a high score). Subsequently, it propagates this score to all the facts extracted by this pattern.

3.3. Applicability to German

In this section, we will examine the general differences between German and English and if the approaches described in section 3.1 could get applied to German texts.

Overall, there are two main input types, POS tagged text (with or without NP-chunks) or dependency parsed text (it must be noted that dependency parser requires POS tagged input, so these systems can use both). Link Grammar will not be listed separately in our consideration, because it can be seen as dependency parsing. Two uncommon systems are SRL-IE, which converts SRL extractions to OIE extractions, and CSD-IE, which relies only on constituent parsing and converts this to a sentence-constituent-identification tree, a special format designed to enable easy extraction of OIE triples.

3.3.1. German versus English

Alphabet Additionally to the 26 Latin based letters in English, German has "ß" and *Umlaute* (ä, ö, ü). This poses no problem because all necessary features are word based.

Gender unlike English nouns, German nouns are either masculine, feminine, or neutral. The article depends on the gender of the noun:

die/eine Sonne, der/ein Mond, das/ein Haus
the/a sun, the/a moon, the/a house

This complicates the construction of correct phrases.

Cases German has four cases which describe a word's function in the sentence: nominative, accusative, dative and genitive. These cases are needed to get the correct meaning of a sentence, because the word order is not as fixed as in English. The article changes depending on the case, too.(Hentschel and Weydt, 2003, pp. 167-190)

Word order English has a specific subject-verb-object order, whilst in German, there are only few rules for the word order. The four cases complement the missing rules and provide the information needed to understand a sentence.

3.3.2. PoS

All of examined systems used language specific POS tags. The main reason for this is that a language specific POS tagset provides more information than a universal tagset, which leads to better results of the OIE system. Petrov, Das, and McDonald (2011) proposed a universal tagset consisting of twelve part-of-speech categories, working for 22 different languages. These tags are very general, for example there is only one tag for all verbs, making it impractical for both German or English OIE. Consider these sentence tagged with Penn Treebank P.O.S. Tags:

Akim/NNP tries/VBZ to/TO jump/VB and/CC stumbles/VBZ
Akim/NNP tries/VBZ to/TO jump/VB and/CC sing/VB

With distinction between VB (Verb, base form) and VBZ (Verb, 3rd person singular present) it would be theoretically possible to distinguish between the first case, where *tries to* is meant for both jump and stumbles, and the

second case, were *sing* is disjointed from *tries to*. In the universal tagset all verbs have the same tag, making those sentences identical, so it would not be possible to correctly identify both relations in both sentences.

PennTreebank vs STTS

In this section, we compare common English POS tags with common German POS tags, to check how different they are and to get an understanding if an OIE system using POS tags as features could possibly achieve similar results as the existing English systems.

A widespread English tagset is the one provided in the PennTreebank (PT) (Marcus, Marcinkiewicz, and Santorini, 1993). It covers 36 POS tags and 12 other tags (for punctuation and currency symbols). For German, Tiger and Negra are two popular treebanks, both using the Stuttgart-Tübingen-Tagset (STTS) (Schiller, Teufel, and Thielen, 1999) (Tiger with small variations). STTS includes 48 POS tags and 6 other tags (for foreign material, punctuation, etc.).

As those tagsets were developed independently for different languages, they obviously differ in structure. Generally, English has a simpler grammatical structure than German, which leads to a smaller number of necessary POS tags. For example, STTS describes 14 different categories of pronouns, while PT includes 4 pronoun categories and categorises some of the pronouns (which, what, that, all, both...) with one of the three available determiner tags. But there are some parts where PT and STTS could have used the same depth of categorisation, but where one of both has a more accurate classification:

- nouns: PT distinguishes between singular and plural, STTS does not
- verbs: PT differentiates between tenses
- adjectives, adverbs: PT also has POS tags for comparative and superlative
- prepositions: STTS distinguishes between preposition, postposition and circumposition

The missing information about nouns should not be an issue for German OIE. In OIE, the aim is to identify subject, verb and object, and it is of

minor importance if subjects or objects are singular or plural. Even with the declaration as plural it is only known that there is more than one, but the exact number is still not known. The situation is similar with adjectives, OIE has basically no interest in the knowledge if we have a basic, comparative or superlative adjective. Postposition (a year *ago*) and circumposition (from now *on*) are very rare in English; for this reason they do not have separate POS tags in PT. They are more common in German, hence the additional tags. This should not impair a German OIE. The last difference is concerning the tense of a verb, which is not completely provided in STTS. The tense is important in facts so that it is clear if something was in the past or is in the present. Even without explicit tags it should still be possible to extract correct facts, since the tense is implicitly included in the written verb. A problem of POS tags for a German OIE is that they do not provide information about cases. As the cases are essential to complement the free word order, important information is missing when relying only on PoS.

3.3.3. Dependency Relations

The dependency relation tagset of data-driven dependency parsers such as Mate Tools[2] or MaltParser[3] depends on the treebank the parser is trained on. Popular German resources are the TIGER and TüBa-D/Z treebanks. For English no large-scale dependency treebank is available, but it is possible to convert constituent-based formalism to dependencies. Surdeanu et al. (2008) describes the algorithm which was used to create dependency labels for the Penn Treebank.

DepPattern (Gamallo, 2015) is rule-based dependency parser, which allows to define grammatical rules for a language which will be used to build a dependency parser. The dependency tags are also part of the grammatical rules which have to be provided for a language. This means that dependency tagset of DepPattern may vary according to the given rules.

The Stanford Parser[4] moved from English specific dependencies to Universal

[2] https://code.google.com/p/mate-tools/
[3] http://www.maltparser.org/
[4] http://nlp.stanford.edu/software/lex-parser.shtml

Stanford Dependencies (USD) (De Marneffe et al., 2014), which can be used to capture any dependency relation between words in any language, without losing to much information. Because some languages have special grammatical relations, USD allows language specific relations.

The USD shows that the dependency relations can be described with one set for many languages, without leading to too general relations. A total of 42 relations are described, which support also the German language. Compared to the Tiger and TüBa-D/Z treebanks, which use 44 and 40 grammatical function labels, it seems that these are similar.

This leads to the conclusion that grammatical relations in German can enable similar OIE results as English systems which work with dependency parses.

3.4. Performance Comparison

We showed in section 3.3.1 that all of the approaches listed in section 3.1 are expected to achieve similar results when implemented for German. For that reason, we will compare the performance of the systems listed in table 3.1 in this section. The approaches of the leading systems will be considered for implementation in our German OIE system. The results published in the papers will be used for comparison. There was no coherent dataset used in all the experiments, and also the opinions how to label extractions as correct or incorrect vary wildly. We will use all the reported evaluation results to infer an overall ranking. For example, if A reported higher F1-measure than B (with identical test conditions for A and B), and B reported higher F1-measure than C (with identical test conditions for B and C), A will be ranked above C. This induces a separation of OIE systems by language.

3.4.1. English

16 of the 20 OIE systems focused on English. ReNoun and Xavier and Lima (2014) will be ignored because they are specialised on some specific relations not captured by other systems. Patty is also disregarded, due to the different

Table 3.3.: Performance comparison of OIE systems. The number in a cell indicates the position of a system among the other systems in the same column. (1 = best system)

	Y. Xu et al., 2013	Bast and Hausmann, 2013	Castella Xavier et al., 2013	Del Corro and Gemulla, 2013	Schmitz et al., 2012	Akbik and Löser, 2012	Christensen, Soderland, Etzioni, et al., 2010	Fader, Soderland, and Etzioni, 2011	Wu and Daniel S Weld, 2010
TextRunner				5			2	4	
WOE_{parse}				3	3			2	1
WOE_{pos}								3	2
ReVerb	2	4	2	4	2	2		1	
SRL-IE							1		
Kraken						1			
OLLIE	2	3		2	1				
ClausIE		2		1					
LSOE			1						
CSD-IE		1							
TK	1								

purpose of the system ((building a taxonomy), so the results can not be compared to other systems.

In table 3.3 the relative ranking of the systems is shown. The system introduced by the authors shown in the column header always got best results, indicated by "1" (first place), while the next best system included in the comparison got a "2", and so on. Although some datasets were reused to support comparability, the configuration of the tested systems, the choice of performance measures and the evaluation were done individually per paper. One example for this is the evaluation dataset created by Fader, Soderland, and Etzioni (2011), which was reused by Akbik and Löser (2012), Del Corro and Gemulla (2013) and Y. Xu et al. (2013). Fader, Soderland, and Etzioni (2011) used the total number of correct extractions as the measure of recall for the corpus. Due to the fact that they did not report the exact number of

extractions of the tested systems, the recall values cannot be compared to by others who use the same dataset. Akbik and Löser (2012) and Del Corro and Gemulla (2013) decided to provide the number of correct extractions and the total number of extractions instead of a recall value. Despite using the same dataset, they report different results for ReVerb. This is probably because of different configurations of the system, but it complicates comparison.

Table 3.3 shows that newer systems almost always outperform previous ones, which is reasonable. Additionally, the order of the systems matches in nearly all reports (only Del Corro and Gemulla (2013) and Fader, Soderland, and Etzioni (2011) state different results for WOEparse and ReVerb).

ReVerb achieves better or similar results than the previous systems TextRunner, WOEparse and WOEpos. Wanderlust and StatSnowball do not appear in the table because they used separate datasets, but the reported results (Akbik and Broß, 2009; Zhu et al., 2009) suggest performances similar to TextRunner and both WOE systems.
LSOE is the newest English OIE system which is still working with shallow features only, and was compared by Castella Xavier et al. (2013) to the older ReVerb system. Castella Xavier et al. (2013) stated only slightly better results in terms of precision for LSOE, while ReVerb yields a far higher recall value. The difference in recall shows that there is a vast number of rules which are hard to describe manually, like LSOE tries to. The missing improvement of the results indicates that OIE with shallow features can not be easily optimised any more. It also explains the fact that nearly all of the newer systems work with deep parsing.

Schmitz et al. (2012) show that OLLIE, which switched from shallow parsing to deep parsing, got 4.4 times more correct extractions than its direct ancestor ReVerb, at a precision of about 0.75. Although 30% of OLLIE's extracted facts which ReVerb misses are contributed by non-verb mediated facts, about 70% were found because of the available deep linguistic information.
Akbik and Löser (2012) described the results of a comparative evaluation of Kraken and ReVerb,and reported higher precision (0.68 versus 0.64) and more extracted facts (572 versus 528) for the dependency parser based Kraken system. These are small improvements compared to OLLIE, but Kraken actually focused on extracting complete facts. And in that category Kraken nearly doubled the number of complete and true facts (0.79 versus

0.43).
The two most promising candidates are ClausIE and CSD-IE. ClausIE was comprehensively compared to four other state-of-the-art systems by Del Corro and Gemulla (2013). The authors used the dataset provided by Fader, Soderland, and Etzioni (2011) and additionally created two new test datasets from New York Times and Wikipedia articles. Their results confirmed existing reports: OLLIE performs better than ReVerb, WOE and TextRunner. They also found out that ClausIE produced about three times more correct facts than its strongest competitor OLLIE, which is a huge boost. But this major improvement in the number of correct extracted facts does not mean that it also extracted three times more information. ClausIE is able to generate multiple propositions for one extracted clause: When OLLIE would extract *[Albert Einstein][died][in Princeton in 1955]*, ClausIE can generate three facts:
[Albert Einstein][died][in Princeton in 1955]
[Albert Einstein][died][in Princeton]
[Albert Einstein][died][in 1955]
The non-redundant number of facts of ClausIE is nearly a third less than the number of facts including redundant ones. This is still better than OLLIE, and, more importantly, it proves that rule based systems with dependency parsing work at least as well as trained systems, without needing many rules (Del Corro and Gemulla (2013) described seven basic clause types).
The strongest competitor to ClausIE is CSD-IE. Bast and Haussmann (2013) compared ReVerb, OLLIE, ClausIE and CSD-IE using the datasets described by Del Corro and Gemulla (2013) (which were also used for the ClausIE comparison). The ranking reported by Del Corro and Gemulla (2013) was affirmed, while the new CSD-IE system got even better results (474 correct facts) than ClausIE (421 correct facts). The good results are established because CSD-IE also extracts is facts (see subsection 4.1.4). This leads to a larger number of smaller facts. *[President Barack Obama][lives][in the White House]* is separated into two facts: *[Barack Obama][lives][in the White House]* and *[Barack Obama][is][President]*.
Y. Xu et al. (2013) evaluated TK, once compared to ReVerb using the ReVerb dataset, and once compared to OLLIE with a new dataset. TK's F-score was 9% higher than ReVerb's. Compared to OLLIE, TK extracted 57% more noun-mediated relations, but did worse with verb-mediated relations, overall, it got an about 20% higher F-score. These are good results, but the improvement does not seem to be as good as ClausIE's or CSD-IE's.

SRL-IE was only compared once to TextRunner by Christensen, Soderland, Etzioni, et al. (2010) who showed that it doubled TextRunner's F1 result. The processing time is however really slow, SRL-IE needed 495 times longer to process the same dataset than TextRunner. We consider SRL-IE's results to be similar to ReVerbs, since they reported roughly the same values.

To conclude, it can be said that dependency parser based systems yield the most promising results for the future, as they enable higher precision and recall than shallow feature based OIE systems. ClausIE and CSD-IE demonstrate how a small collection of rules easily competes with trained systems. Unlike ClausIE, CSD-IE refrains from using a dependency parse tree, and instead, creates its own sentence-constituent-identification tree out of a constituent tree, which enables simple derivation of facts. Both systems accomplish state-of-the-art results, which makes both approaches very promising for implementation in a German OIE system.

3.4.2. Other Languages

Little work has been done in OIE for other languages than English. The most interesting work here has been done by Gamallo, Garcia, and Fernández-Lanza (2012) and Gamallo and Garcia (2015), who addressed multilingual (English and Romance languages) OIE. Both DepOE (Gamallo, Garcia, and Fernández-Lanza, 2012) and its successor ArgOE (Gamallo and Garcia, 2015) utilise dependency parsing to extract facts in multiple languages. They use DepPattern (Gamallo Otero and González López, 2011), which can generate dependency parsers from DepPattern grammars. Basic grammars are provided for English and Romance languages, and the creator mindfully used the same dependency labels for all of them. Gamallo, Garcia, and Fernández-Lanza (2012) reported that DepOE achieved higher precision and a slightlylower recall than ReVerb in English. These mediocre results originate mostly from DepPatterns parsing errors. Also the successor ArgOE was clearly outperformed by ClausIE in terms of precision and recall (Gamallo and Garcia, 2015), again the given reason was DepPatterns parsing errors.

Two other single language OIE systems are ExtraHech (Zhila and Gelbukh, 2013) for Spanish and SCOERE, which targets Chinese OIE. Zhila and

Gelbukh (2013) reported that "ExtrHech performs at the precision and recall levels comparable with the state-of-the-art systems for English based on similar approach". Wang, Li, and Huang (2014) evaluated SCOERE only on a Chinese dataset consisting of news articles, and reported 74.3% recall and 72.2% precision.

Three of the non-English OIE systems work with dependency parsed sentences and one with POS tagged sentences. Both design decisions worked, it seems that SCOERE with POS worked as well as ArgOE with dependencies, because of the mediocre performance of the used dependency parser. Only the Chinese system trained the extractor, the others created rules manually. No reason was given why manual rules were chosen above training data, but with the reported results we can assume that these constitute manageable effort with better outcome.

3.4.3. Dependencies or Constituents

We found out in this section that dependency relations (advocated by ClausIE) and constituent relations (advocated by CSD-IE) both enable state-of-the-art OIE. While dependency relations provide enough information to directly apply rules to the sentence and extract relations from it, constituent relations have to be postprocessed. As Kübler, Hinrichs, and Maier (2006) already said: "obtaining the correct constituent structure for a German sentence will often not be sufficient for determining its intended meaning.". Therefore, CSD-IE uses rule based conversion to transform it into a CSD-tree, which suits better for relation extraction. This process reminds of the rule based conversion of a constituent tree to a dependency tree and speaks in favour of directly using dependency relations. Another reason is that opposed to English parsers, German dependency parsers perform better than Constituent parsers (Kübler and Prokic, 2006). This leads to the conclusion that CSD-IE's approach would not work as well in German. Also the fact that most of the non-English OIE systems rely on dependencies, and none on constituent representation, supports the ClausIE's approach. Based on this knowledge we chose to implement a German OIE system with hand-crafted rules for dependency relations. In section 3.5 we will investigate which state-of-the-art dependency parsers for German are available.

3.5. Dependency Parser

In this section, we will give the reader a short overview of dependency parsers, more precisely existing techniques, progress due to events promoting (multilingual) dependency parsing and specific parsers available which work with German sentences.

3.5.1. Techniques

Data-driven

Transition-based dependency parsers gradually build the tree by applying a sequence of transition actions (Yamada and Matsumoto, 2003; Nivre, 2003). The sum of the scores of these actions is equal to the score of the tree. The parser aims to find the sequence which results in the highest score and a legal tree. To find the optimal action sequence a greedy algorithm is often used, with a typically linear or quadratic complexity (Hall and Nivre, 2008; Attardi, 2006).
Graph-based systems try to score a dependency tree by factoring the scores of the tree's subgraphs. The methods how the score is calculated differ: they can, for example, be restricted to linear classifiers or joint probabilities. A widely used form of graph-based dependency parsing is called arc-factored parsing, where the single dependency arcs (edges) get parametrised. The the worst case complexity is $O(n^2)$ for for non-projective algorithms and is $O(n^3)$ for for projective algorithms (Kübler, McDonald, and Nivre, 2009).

Grammar-based

Kübler, McDonald, and Nivre (2009) distinguished between two types of grammar-based parsing methods, the constituent-based and constraint-based method. The constituent-based method takes the output of a constituent parser and uses production rules to transform it into dependencies. The second approach uses constraints, generally written manually, to generate the dependency structure. A constraint restricts the possible heads of

a word and the possible dependencies between words, for example, that a countable noun requires a determiner. Since the use of hard constraints (which must be satisfied) is too strict, weighted constrained dependency grammar was developed (WCDG), where each constraint is assigned a weight. This way the relative importance of a constraint can be described (Kübler, McDonald, and Nivre, 2009).

3.5.2. Events promoting dependency parsing

Since the Conference on Computational Natural Language Learning (CoNLL[5]) has promoted multilingual dependency parsing and provided resources for this, much progress has been made in this area and the number of freely available dependency parsers has increased. The published results of the parsers which participated in various tasks in CoNLL-X, CoNLL 2007, CoNLL 2008 and CoNLL 2009 enable fair comparison, also for dependency parsers which did not participate directly, but which could also use the provided data to compare their performance with others. One of the best multilingual dependency parsers in both CoNLL-X and CoNLL 2007 was MaltParser (see subsection 3.5.4), which underwent several updates since then and is still state-of-the-art. The Mate Tools dependency parser emerged from an approach which scored best for the dependency parsing task in German in CoNLL 2009.

A more recent event, the SANCL 2012 shared task[6](Petrov and McDonald, 2012), focused on parsing texts from unedited domains, such as blogs, discussion forums or consumer reviews. As Open Information Extraction aims to extract all sources from the web, this is especially interesting, but at the moment this task addresses only English texts.

SemEval 2014/2015[7](Oepen et al., 2014) concentrated on broad coverage semantic dependency parsing, also currently only for English systems. Semantic dependency parsing aims to get a more direct analysis of "who

[5] http://www.conll.org/
[6] https://sites.google.com/site/sancl2012/home/shared-task
[7] http://alt.qcri.org/semeval2014/,http://alt.qcri.org/semeval2015/

did what to whom", moving away from tree representation with only one root, because a node can be the argument of multiple predicates.

A newly presented comparative analysis of ten leading (English) statistical dependency parsers on a multi-genre corpus by Choi, J. Tetreault, and Stent (2015) concluded that the Mate-tools dependency parser still achieves overall the highest scores (Labelled attachment score: 90.34), although competition has increased and new multilingual dependency parsers like Yara (Rasooli and J. R. Tetreault, 2015)[8] or ClearNLP (Choi and McCallum, 2013)[9] (which is based on the algorithm of MaltParser) are very close in performance. The parsing speed without greedy algorithm is at 30 sentences per seconds for mate-tools, 18 for Yara and 72 for ClearNLP. This is about ten times slower compared to greedy parsing, but greedy parsing also reduces accuracy by some points. It has to be considered that these results originate from English texts, and cannot be directly applied to German.

Lavelli (2014) compared the results obtained by the participants in the EVALITA 2014[10] Dependency Parsing Task, which targets Italian texts. Mate-tools and MaltParser are again among the top parsers, with an accuracy of around 87%.

3.5.3. Mate Tools

Mate Tools[11] offer modules for lemmatisation, part-of-speech tagging, morphologic tagging, dependency parsing, and semantic role labelling of a sentence. All tools in this pipeline are language independent. The provided input sentences have to be tokenised. Two different dependency parsers are available, a graph-based and a transition-based dependency parser.

The **graph-based parser** is an improved version of the state-of-the-art dependency parser developed by Bohnet (2010). The original version (Bohnet, 2009) performed very well in the Conll2009 Shared Tasks[12]. In the syntactic

[8] https://github.com/yahoo/YaraParser
[9] https://clearnlp.wikispaces.com/depParser
[10] http://www.evalita.it/2014/
[11] https://code.google.com/archive/p/mate-tools/
[12] https://ufal.mff.cuni.cz/conll2009-st/

dependency parsing task, this parser could reach the second place with an accuracy of 85.68 on average, and additionally, it scored highest in English and, more importantly, German. Also the out-of-domain data task was dominated with an average accuracy of 78.79. The first implementation had a relatively slow speed, so Bohnet (2010) resolved this issue by using a passive-aggressive perceptron algorithm as a Hash Kernel. This improved parsing times and additionally yielded higher accuracy, because it takes into account features of negative examples created during the training. As current computers employ multi-core processors, parallel feature extraction and parsing was implemented, too. The Hash Kernel alone could boost the parsing time by about 350%, multi-threading further increased it by a factor of 4.6 with 4 CPU cores and hyper threading. This reduced the required time to parse one sentence from 1235 to 77 milliseconds in average[13].

The **transition-based parser** implements the techniques described by Bohnet and Kuhn (2012) and Bohnet and Nivre (2012). It uses the transition-based algorithm at the top level which has the advantageous property of having a quadratic complexity in the worst case. Additionally, some ideas from the graph based-approach are used, instead of the best-scoring histories of transitions, the k best-scoring are stored (Johansson and Nugues, 2006). Since new information during the parsing process may give new insights, the scores of the transitions in the history, which could not be decided upon previously, are recalculated after every new word (Bohnet and Kuhn, 2012). The accuracy of the first approach (Bohnet and Kuhn, 2012) was a bit lower than the older graph-based parser (Bohnet, 2010), though this could be improved by Bohnet and Nivre (2012), who presented a joint part-of-speech tagging and labelled dependency parsing approach. The combination could obtain 89.05 for German in the syntactic dependency parsing task of CoNLL 2009, which is 1.57 points better than Bohnet (2010).

Motivated by the results of Bohnet and Nivre (2012), Bohnet, Nivre, et al. (2013) explored various options of integrating morphological features into the model. It was shown that joint prediction models, rule-based lexical constraints, and distributional word clusters improve accuracy for richly inflected languages.

[13]The test system used a Intel Nehalem i7 CPU 3.33 GHz, overclocked to 3.46 GHz

The **German lemmatiser** has an accuracy of 98.28% and the **German POS tagger** achieves 97.23% (Björkelund et al., 2010).

3.5.4. MaltParser

MaltParser (Hall and Nivre, 2008)[14] is a freely available multilingual dependency parser, which was one of the top performing systems in the CoNLL 2006 and 2007 shared tasks (Buchholz and Marsi, 2006; Nilsson, Riedel, and Yuret, 2007). MaltParser belongs to the transition-based parsers and uses a greedy parsing algorithm developed by **nivre2006inductive** (Hall and Nivre, 2008) reported for the improved version label attachment scores of 90.8% for TIGER and 88.46% for Tüba-D/Z treebank, which is better than the best system in the CoNLL-X shared task obtained for German.

3.5.5. ParZu

As past conferences featured either English or multi-lingual dependency parsers, ParZu (Sennrich, Schneider, et al., 2009) could not participate in these events, because it is German only. As the monolingualism allows for an interesting approach, we will describe it here. Sennrich, Schneider, et al. (2009) created ParZu[15] by modifying the English Pro3Gres parser (Schneider, Hess, and Merlo, 2008) and adapting it to German. It uses a hybrid architecture which combines a manually written functional dependency grammar with statistical lexical disambiguation obtained from the TüBa-D/Z corpus. The first version achieved results similar to MaltParser with a parsing speed of 10.9 sentences per second (with gold morphological and gold POS tags). Sennrich, Volk, and Schneider (2013) discussed various ways to improve ParZu, showing that the integration of morphology tools yields a very accurate model. ParZu supports the morphology tools Zmorge (Sennrich and Kunz, 2014), GERTWOL[16] and Morphisto (Zielinski, Simon, and Wittl, 2009)[17], which all lead to a parsing precision around 89.8% (using

[14] http://www.maltparser.org/
[15] https://github.com/rsennrich/ParZu
[16] http://www2.lingsoft.fi/doc/gertwol/
[17] https://code.google.com/archive/p/morphisto/

gold POS tags).

3.5.6. CDGParser

CDGParser (Foth, Daum, and Menzel, 2004) is grammar based, it uses rules as declarative, defeasible constraints specifically for German grammar, which makes it largely independent of text type and domain. Foth, Daum, and Menzel (2004) reported a labelled attachment score of 87.0%, which was very competitive, but due to the lack of recent updates we consider this parser inferior to current state-of-the-art dependency parsers.

3.5.7. Stanford and Berkeley Parser

The Stanford and Berkeley Parser are both not capable to generate German dependency parses at the moment, we mention them here briefly, because they are well known and open source, and especially the Stanford parser is used in some of English OIE systems, e.g. ClausIE. Both support multilingual constituent parsing, and the Stanford Parser uses rule based constituent transformation to generate the dependency structure, but these grammatical rules are language specific and not implemented for German yet.

3.6. Evaluation

All of the observed OIE systems (3.1) were manually evaluated by human judges. In the following subsections, 3.6.1 and 3.6.2, we will describe which measures were used and how the datasets were build. We will use this information to create a proper German dataset and decide how to evaluate the extractions (5).

3.6.1. Measures

Here we give an, overview of measures used in the evaluation of the systems displayed in table 3.1.

Correctness was used in every evaluation, since it is the most important measure. Incorrect extractions are useless for further usage, so it is necessary to know their proportion within all facts. It was not always stated how the correctness was asserted. Banko, M. J. Cafarella, et al. (2007) declared that a fact which is "consistent with the truth value of the sentence from which it was extracted" is correct. Additionally, correct triples have to be well formed: [[entity][relation][entity]). A similar definition was used by Schmitz et al. (2012). Del Corro and Gemulla (2013) on the other hand, declared that the context was ignored when evaluating the correctness. For example, the extraction *[I][can see][very well]* from the sentence "I can see very well when I wear glasses." was seen as correct. Castella Xavier et al. (2013) was more restrictive as they classified uninformative (*[Obama][president of][State]*) or incoherent triples (*[Obama][president company][America]*) as incorrect. Gamallo and Garcia (2015) also labelled overly specific extractions (triples including e.g. named entities or very long phrases) as incorrect.

Completeness While others said that a fact has to be complete to be correct (which obviates an extra measure for this), or just ignored completeness, Akbik and Löser (2012) decided to evaluate completeness independently of correctness. The extraction *[Elvis][moved to][Memphis]* from the sentence "Elvis moved to Memphis in 1948" was seen as correct, but incomplete, because of crucial information loss.

Minimality As Bast and Haussmann (2013) made Minimality ("can the extracted triple be further decomposed into smaller meaningful triples") a requirement for their extracted facts, they also evaluated if this requirement was met for each extracted triple.

Concrete/Abstract Banko, M. J. Cafarella, et al. (2007) examined whether a fact is concrete or abstract. They defined that a fact is concrete when "the truth of the tuple is grounded in particular entities". A given example for this was *[Tesla][invented][coil transformer]*. All other (underspecified) facts, such as *[Einstein][derived][theory]*, were labelled as abstract. This distinction was made due to the different applications of those facts; concrete facts are more useful for question answering, while abstract facts can, for example, be used for ontology learning.

Precision This measure is the fraction of extracted facts which are correct, and was often provided because it is a well known measure. To allow fair comparison of precision values, same evaluation conditions (dataset, definition for correctness...) have to be ensured.

Recall Recall is the fraction of facts which got extracted. Due to the circumstance that it is difficult to say how many facts exist in total, only a few teams tried to provide a recall value. Two approaches to identify the total number of facts where found. One was to manually label all possible relations in the dataset (Akbik and Broß, 2009; Wu and Daniel S Weld, 2010; Fader, Soderland, and Etzioni, 2011). Here, it is necessary to argue what counts as a fact. A simpler approach is to just use the total number of extractions (union, when multiple systems were evaluated). This pseudo recall is seen rather off the true recall value, since most of the systems do not achieve a high recall value.

3.6.2. Datasets

The decision how to compose the evaluation dataset is crucial, it should reflect real world data from where the system will be applied. Open Information Extraction aims to extract facts from the Web, so most of the used datasets consist of sentences randomly sampled from the Web. Since news pages and Wikipedia are important sources of information, some datasets were also created out of those domains. The size of the used datasets was

usually around 500 sentences, because any larger sets would lead to excessive manual labour. Most of the datasets consist of only well formed sentences. This is acceptable, because sentences from reputable sources tend to be well formed. Akbik and Broß (2009) only used sentences in their evaluation dataset which contained entities which have their own Wikipedia page. Such restrictions are too specific and impede comparison with other systems.

3.6.3. Automatic Evaluation

When evaluation has to be done manually, it often involves a lot of manual labour and there are often inconsistencies between different evaluations. Hence an interesting research field is the automatic evaluation of OIE systems. The first to tackle this problem wereBronzi et al. (2012). They proposed an evaluation framework for automatic evaluation of relation extraction systems based on realistic-sized corpora. For this, they used an existing database (FreeBase) and the Web as ground truth. They determined that "A fact is said to be correct if (1) we can find the fact in the database or (2) we can detect a statistically significant association between e1, e2 and r on the web.". Recall is a difficult measure for OIE because the total number of facts is unknown, it depends on the definition what a fact is. Bronzi et al. (2012) tried to estimate the number of facts, by inferring the total number of facts from the number of distinct facts generated by the system and in the database. They showed that their method provides fair evaluation of OIE systems with corpora containing over a million of documents. Automatic evaluation alleviates the development of OIE systems, but as long as such a framework is not freely available, it has to be done manually.

4. GerIE

This chapter presents GerIE, the first German Open Information Extraction system. As discussed in subsection 3.4.3, we chose to create hand-crafted rules working on dependency parsed sentences. For parsing the sentences we selected the most current version (*anna 3.61*) of Mate Tools (discussed in 3.5), which can carry out lemmatisation, part-of-speech tagging, morphological tagging and dependency parsing. The provided models were trained on the full German Tiger corpus[1]. This means that all extraction patterns were specifically created for the tagsets that come with this corpus[2]. Any parsers trained on the tiger corpus can therefore be used by GerIE. In section 3.2, we described several principles which were applied by other OIE systems. We decided to adopt *Minimality* because it improves the quality of the extracted facts and *separation of detection and representation*, because this helps to enforce minimality and allows for easier changes and customisation, too. *Levels of granularity* was not seen as important because GerIE is only a prototype and no "substantial postprocessing" is planned, and if necessary, additional output information could still be added later on with very low effort. *Separation of relation detection and extraction* was also discarded as these two parts are tightly bounded; the moment a proper relation pattern is found in a sentence, it should be extracted. The idea of adding *context analysis* is very useful, but we decided not to implement it in GerIE, as this can be seen as additional task. It was still regarded in the design of GerIE, so it can smoothly be added at a later time. The *confidence score* would be an additional module which does not influence the structure of the system. Although it can also improve the quality of the results, we postponed this for later work. There are two open tasks, the first is to give

[1] http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.en.html

[2] http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/TIGERCorpus/annotation/tiger_introduction.pdf

a confidence score to each extraction pattern and the second is to rate the final extractions.

To achieve a high recall, we decided to try to capture as much relation types a possible. Therefore we extract verb-mediated relations, noun-mediated relations, is relations and has relations, which are listed in section 4.1. Since adjective-mediated relations and adjective noun pairs are covered by is relations, we do not treat them separately. Noun compounds are also not considered, because we decided that the approach described by Xavier and Lima (2014) can be substituted by directly extracting relations from the external resource. We also contributed a new relation type, the proposition relation. An example for this is *[New York][in][USA]* or *[Harry Potter][from][Hogwarts]*.

Separation of detection and representation allows to create output in an arbitrary format, we support the generation of n-ary propositions. As it is difficult to get the correct word order of multiple objects, we exclude the generation of triples. In section 4.2 we explain how we made out the patterns implemented in GerIE, section 4.3 describes the components of GerIE.

4.1. Relation Types

Various grammatical structures exist which express diverse relations. By analysing existing OIE systems we found six categories of relations which were of interest for those systems. This section will give an overview of these categories.

4.1.1. Verb-mediated Relations

Verb-mediated relations are represented in a subject-verb-object structure, therefore, every common sentence includes at least one of these relations: *Akim hates raisins.* Especially with dependency parsing, these are straightforward to extract. Until OLLIE (Schmitz et al., 2012), all of the OIE systems concentrated only on this type, Etzioni, Banko, et al. (2008) showed that

most of the explicitly expressed relations in a sentence are verb-mediated relations.

4.1.2. Noun-mediated Relations

Schmitz et al. (2012) were the first to expand the syntactic scope of relation phrases to additionally cover relations mediated by other word classes than verbs, but which are still explicitly constituted in a sentence: noun-mediated relations (described here) and adjective-mediated relations (described in subsection 4.1.3).

"Obama, the president of the US", "Obama, the US president", "US President Obama" → *[Obama][president (of)][(the) US]*

A noun-mediated relation composites of two entities and a noun expressing the relations between those entities.

4.1.3. Adjective-mediated Relations

The only system which extracts relations mediated by adjectives is OLLIE. The idea here is to capture relations between two entities: "the great singer Michael Jackson" → *[Michael Jackson][great][singer]*. This type only captures relations when an adjective is present, which is not always the case, like in the phrase "the singer Michael Jackson". A better approach for this are is relations, described in subsection 4.1.4), which can handle both *[Michael Jackson][is][great singer]* and *[Michael Jackson][is][singer]*, and therefore provide a more generic solution.

4.1.4. Is Relations

Is relations belong to the implicit relations, hence there is no explicit verb "is", but it can easily be deduced from the context: "the great singer Michael Jackson" → *[Michael Jackson][is][(great) singer]*.
This kind of relation can appear between two entities, one of those may be

a named entity, like in the example above. If both are just common nouns, the fact is of more general nature like *[mouse][is][mammal]* or *[wolf][is][pack animal]*.

4.1.5. Has Relations

Similar to is relations, has relations are implicitly expressed in the sentence. *"The president of the USA died." [USA][has][president]*
These facts can be of very abstract character, but those are still useful to get knowledge of the world's structure: *They are useful to get knowledge of the world's structure. [world][has][structure]*

4.1.6. Noun Compounds and Adjective Noun Pairs

Xavier and Lima (2014) focused on relations implicitly expressed in noun compounds (NCs) and adjective noun pairs (ANs). An example given by Xavier and Lima (2014) for an AN was "raw food", which can be interpreted by the relation (food, that is, raw) and for the NC "glass vase" the relation (vase, made of, glass). For ANs, Xavier and Lima (2014) assumed that adjectives describe a certain quality of this noun, therefore the relations *"that is"* was always taken for those compounds. This is actually very similar to is facts, which means that is relations also appear between adjectives and nouns. The relations in NCs depend, as noticeable in the examples, on the entities, therefore, the author proposed the approach to get a proper synset from an external resource. WordNet[3] provides the information that "oil industry" is in the same synset as "industry that produces and delivers oil". The propositions gained from ANs and NCs are again of very general nature.

[3] http://wordnet.princeton.edu/

4.2. Pattern creation

To find possible patterns, we first analysed the trial dataset provided in the CoNLL-2009 Shared Task[4]. It contains 400 sentences which are labelled with gold POS tags and gold dependency tags and heads. Only declarative sentences were used which contained at least one verb. These gold tags ensured that we did not create patterns based on a erroneous dependency structure. We also added our own sentences when a special word order came to our mind which was not covered in the trial dataset, examples of this are shown in table 4.1. We used mate-tools to parse our custom sentences and manually sorted out incorrectly parsed sentences.

Table 4.1.: Examples of custom sentences to complement the CoNLL-2009 trial dataset

Er wird sie singen hören wollen.
Er will sie singen hören.
Er hört sie singen.
Die Kanzlerin Deutschlands, Angela Merkel, ...
Die Kanzlerin von Deutschland, Angela Merkel, ...
Deutschlands Kanzlerin, Angela Merkel, ...
Die Kanzlerin des Landes Deutschland, Angela Merkel, ...

4.3. System Architecture

As displayed in figure 4.1, GerIE is build as a pipeline with several steps: Preprocessing, extraction, postprocessing and proposition generation. As input, sentences in dependency-tree format are expected. For the labels of the edges between the nodes and the POS tags of the nodes the annotation scheme of the TIGER treebank [5] is required. The output are propositions as plain text.

[4] https://ufal.mff.cuni.cz/CoNLL2009-st/trial-data.html

[5] http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.en.html

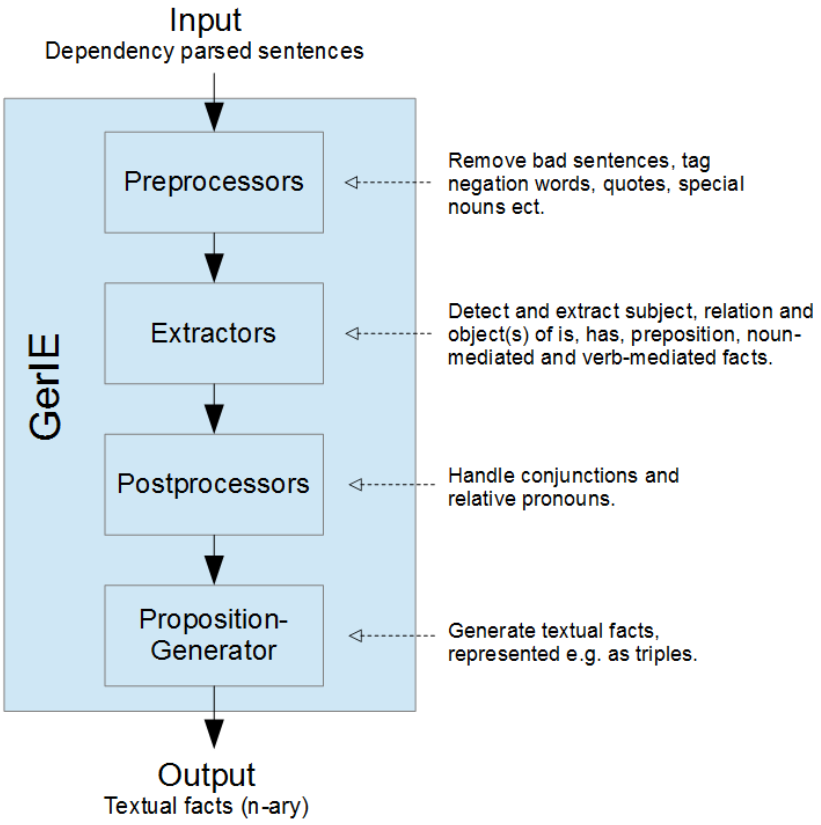


Figure 4.1.: Architecture of GerIE

4.3.1. Preprocessor

The first step is the preprocessing of the received sentences. A Preprocessor takes the dependency-trees and may perform any actions on them. Four Preprocessors are implemented, the *Bad Sentence Filter*, the *Quotes Tagger*, the *Noun Tagger* and the *Negation Word Tagger*.

The Bad Sentence Filter removes interrogative clauses because they are likely to not contain any facts: *"Does alien life exist?"*. Additionally, sentences which do not have a verb as root element in their dependency-tree structure are considered malformed or uninformative and removed: *On the contrary.*

The Quotes Tagger was created to prevent fact extraction from direct speech by assigning special POS tags. The notion behind this is, that direct speech is most probably a personal opinion which cannot be seen as fact: *"Kevin says: 'Alien life exists.'"*. The sentence can still get processed, only the extraction of facts in between the quotes is prevented.

The Noun Tagger's purpose is to divide nouns into more specific categories, as this is sometimes too inaccurate. At the moment, it additionally tags nouns which are numbers ("thousand"), quantities ("handful") or units ("metre, kilogramme"), because these types are important to know for the Is Fact Extractor. This is done simply by comparing the words in a sentence with a gazetteer list (appendix A.2), containing the names of units, numbers, etc.

The Negation Word Tagger also marks words which negate a fact, such as not, no, nobody, with help of a list of known negation words (appendix A.1). These words are essential in a proposition, as they completely change its meaning.

Preprocessors can be added to the pipeline, the user could use the previous mentioned ones or create new ones. This is an optional step which is designed to improve adaptability and accuracy.

4.3.2. Extractor

An Extractor utilises the dependency structure of sentences to detect and extract facts. Five relation types were identified, and for each of these an Extractor was implemented. The selection of words is not done here, a fact just contains multiple Dependency-Subtrees, which represent the subject, the relation, and the objects. So the whole task of an Extractor is to declare a node of a dependency-tree as subject, one node as relation and zero, one or more nodes as objects. This separation allows to add only those extractors to the pipeline which are required. If someone wants to extract a new kind of relation, a new extractor can easily be created and added.

Is Fact Extractor

The Is Fact Extractor captures is relations between entities. X is Y, where X is a common or proper noun, and Y is a common noun: *[Obama][is][president]*, *[lion][is][predator]*.

Altogether, four patterns in the dependency structure were found which express is relations, displayed in table 4.2.

Table 4.2.: Patterns for is fact extraction

	Pattern	**Example Phrases**	**Fact**
$i_1(42):$	$NN \xleftarrow{NK} \text{N}$	Präsident Obama	[Obama][ist][Präsident]
$i_2(19):$	$NN \xleftrightarrow{APP} \text{N}$	der Präsident, Obama Obama, Präsident der Vereinigten Staaten der König des Dschungels, der Löwe	[Obama][ist][Präsident] [Obama][ist][Präsident] [Löwe][ist][König]
$i_3(0):$	$NN \xrightarrow{AG} \text{PIS} \xleftarrow{APP} N$	Obama, einer der Präsidenten	[Obama][ist][Präsident]
$i_4(23):$	$NN \xleftrightarrow{SB\|PD} \text{V} \xleftrightarrow{SB\|PD} N$	Obama ist Präsident	[Obama][ist][Präsident]

Extracted is facts help to make other possible facts minimal. For example, in the sentence *"Präsident Obama lebt seit 54 Jahre."* (President Obama lives for 54 years.) we extract [Obama][ist][Präsident] ([Obama][is][president]).

We don't need to include this information in further facts: [~~Präsident~~ Obama][lebt][seit 54 Jahre] ([~~President~~ Obama][lived][for 54 years]).

i_1 was with 42 correct occurrences the most prominent in the CoNLL dataset. It is also the is pattern with the most (20) incorrect facts there, as it handles following phrases wrongly:

- Nouns representing numbers or quantities, such as *ein Haufen Menschen (a bunch of people), Millionen Menschen (millions of people)* : ~~[Menschen][ist][Millionen]~~
- Phrases where the preposition is not explicitly present, like *Ende März (end of march), Paragraph 129a Strafgesetzbuch (paragraph 129a criminal code), Richtung Wien (towards Vienna)*: ~~[Wien][ist][Richtung]~~
- Nouns representing units, for example *7000 Quadratmeter Büros (7000 square metres office)*: ~~[Büro][ist][Quadratmeter]~~

We eliminated the first and the third problem, which constituted 14 of the 20 false extractions, by assigning a special POS tag to those nouns in the preprocessing step (see NounTagger in subsection 4.3.1). We did not find a simple way to treat the missing prepositions.

i_2 and i_4 appeared 19 and 23 times in the dataset. Both always represented a correct relation, although i_2 is hard to interpret when both entities are nouns: *eine Einrichtung, das Außenministerium* (an institution, the department of state) has identical labels to *das Außenministerium, eine Einrichtung* (the department of state, an institution), so we don't know which noun is the entity and which is the super-entity in the is relation. We decided to take the first entity as super-entity, because this word order occurs more often in the dataset.

i_3 did not occur in the CoNLL dataset, but we considered it a useful pattern which is not uncommon in German sentences. A possible problem here is that the *PIS* (substituting indefinite pronoun) could also be negating, like *keiner* (nobody) or *niemand* (nobody). As negation words are always important to know, we implemented a Negation Word Tagger (see 4.3.1), which tags negation words. These words will then get included in the final propositions.

Has Fact Extractor

The Has Fact Extractor detects has relations in the dependency-tree. Five patterns were identified, displayed in table 4.3, which cover 240 instances in the dataset. We decided that verbs used as nouns (*"... im Wohnen in den Großstädten..."* (living in the big cities)) lead to too abstract facts ([Großstadt][hat][Wohnen] ([big city][has][living])), thus, we excluded these. The preposition patterns $p_1 and p_2$, shown in table 4.4, were initially used to extract has facts, but as the preposition contains important information in these relations, we decided to keep them and created a new relation type. For example *[Amerika][hat][New York] ([America][has][New York])* is not entirely wrong, but *[New York][in][Amerika] ([New York][in][America])* makes much more sense.

The most common relation is the one extracted by h_1, 144 instances were identified. h_2, h_3 and h_4 occurred respectively 39, 2 and 18 times. Many of the has facts (about 85%, those which did not involve a named entity) were abstract, they often describe very general rules:
[child][has][death], *[month][has][imports]*, *[Ausschwitz][has][shadows]*. However, they represent correct knowledge about the world which is still useful for ontology learning, common sense knowledge acquisition and other applications. h_3 identifies an attributive possessive pronoun as first entity, which has to be replaced by the real entity it refers to. This task can be solved with Co-reference resolution. h_5 is a special case, as in German compounds in hyphenated form often contain has facts. We did not find a way to exclude compounds which should not get extracted, but we found that only 8 out of the 39 occurrences in the CoNLL 2009 trial dataset were false has relations, such as *"Hobby-Kicker" (amateur soccer player)* $\rightarrow$ *"[Hobby][hat][Kicker]"* (*[amateur][has][soccer player]*).

Preposition Fact Extractor

The Preposition Fact Extractor identifies relations mediated by prepositions: *[New York][in][America]*. These facts were found to be too inaccurate when handled as has facts, therefore, the preposition is included here. Another difference between preposition facts and has facts is that in has

Table 4.3.: Patterns for has fact extraction

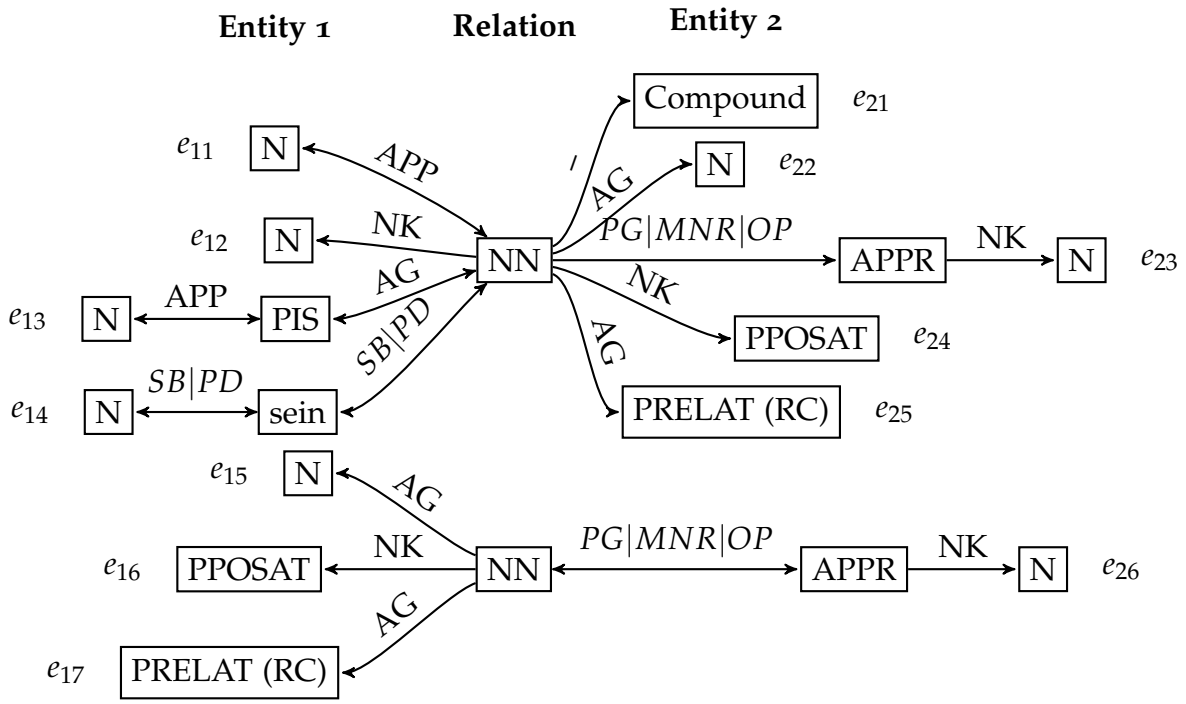
	Pattern	Example Phrases	Fact
$h_1(144)$:	$NN \xleftarrow{AG} \text{N}$	der Kanzler Deutschlands Kanzler des Landes Deutschland Deutschlands Kanzler	[Deutschland][hat][Kanzler]
$h_2(39)$:	$NN \xleftarrow{NK} PPOSAT$	Franz trifft seinen Bruder Karl Perot mit seinem Befehlston Perot sein Vermögen	[Franz][hat][Bruder] [Perot][hat][Befehlston] [Perot][hat][Vermögen]
$h_3(2)$:	$(RC)NN \xleftarrow{AG} PRELAT$	Franz, dessen Bruder	[Franz][hat][Bruder]
$h_4(18)$:	$NN \xleftarrow{PG} \text{APPR} \xleftarrow{NK} N$	der Kanzler von Deutschland	[Deutschland][hat][Kanzler]
$h_5(31)$:	Compound Word	Google-Chef	[Google][hat][Chef]

facts, the second entity is always a common noun. Preposition facts allow the second entity to be a proper noun. The patterns p_1 and p_2 capture 166 relations mediated by prepositions in the dataset. Most of these facts were of abstract nature, for example *[Telekommunikation][mit][Telefon]* (*[telecommunication][by][telephone]*) or *[Regierung][in][Washington]* (*[government][in][Washington]*), *[Anteil][an][Unternehmen]* (*[stake][in][company]*), only those involving two named entities (2 occurrences) were concrete facts. When the first entity is a named entity, the information can be removed from other facts to ensure minimality: *...das Unternehmen National City ~~in Ohio~~...* (...the company National City ~~in Ohio~~...)

Table 4.4.: Patterns for preposition fact extraction

	Pattern	Example Phrases	Fact
$p_1(132)$:	$N \xleftarrow{MNR} \text{APPR} \xleftarrow{NK} N$	Obama aus New York New York in Amerika Krieg gegen Saddam Hussein	[Obama][aus][New York] [New York][in][Amerika] [Krieg][gegen][Saddam Hussein]
$p_2(34)$:	$N \xleftarrow{OP} \text{APPR} \xleftarrow{NK} N$	Mangel an Erfahrung	[Mangel][an][Erfahrung]

Table 4.5.: Patterns for noun-mediated fact extraction



Nodes are POS tags and edges are dependency labels, the direction of the arrows shows which end can be the head.
A relation is found when a path from the left to the right matches a part of the dependency tree of a sentence.

Noun-Mediated-Fact Extractor

The Noun-mediated Fact Extractor extracts noun-mediated facts, which are more specific than is, has or preposition facts. Here, the relation is mediated by a noun phrase: [New York][Stadt in][Amerika] ([New York][city in][America]). Each common noun is a possible mediator between two entities, so every time the extractor encounters a common noun in the dependency tree, it tries to find a left entity and a right entity for it. The patterns to detect both entities are shown in table 4.5. There are multiple possible combinations of patterns for the first entity and patterns for the second entity, both mediated by the same common noun. We displayed each pattern only once, a final extraction pattern is one path from the left (*Entity 1*) to the right (*Entity 2*), over the middle point NN (*Relation*).

These patterns cover 43 noun-mediated relations in the CoNLL 2009 trial dataset. This includes five concrete relations with two named entities, such as *[Karl Polacek][früherer Führer von][FAP]* ([Karl Polacek][former leader of][FAP]), and 38 non-concrete relations, for example *[Indien][Potential als][Markt]* ([India][potential as][market]), *[soziales Ungleichgewicht][Phänomen in][großstädtischem Ballungsraum] ([social imbalance][phenomenon in][megalopolis])* or *[Nahrungsmittelsicherheit][Voraussetzung für][Stabilität]* ([food safety][requirement for][stability]).

e_{21} (Compound) is a special pattern which extracts entities from part of the relation. Here, the relation is a compound with a hyphen (Google-CEO), and the first part is taken as entity 2 and the second as the actual relation. All of the compounds are simply tagged as nouns in the tiger corpus, which makes it hard to distinguish between an entity-relation compound and others (for example *Cyber-Diebstahl* (cyber theft)). Further analysis of the two compound words would be necessary. We encountered three occurrences of this type in the dataset, all separable (*"Landesbank-Chef Hans Fahning"* $\rightarrow$ *[Hans Fahning][Chef von][Landesbank]* ([Hans Fahning][boss of][regional state bank])), hence we did not make any constraints here.

e_{14} is another special pattern, which finds entities in relations explicitly written with a form of the word *sein* (to be), such as *"Obama ist Präsident von Amerika." (Obama is the president of America.)*

As recognizable in table 4.5, *Entity 2* has either the second case (Genitive):

e_{22} : Amerikas Präsident Barack Obama (America's president Barack Obama)
[Obama][Präsident von][Amerika] ([Obama][president of][America])
e_{25} : Amerika, dessen Präsident Barack Obama (America, whose president Barack Obama)
[Obama][Präsident von][Amerika] ([Obama][president of][America]

or is referenced by an attributive possessive pronoun:

e_{24} : Obama füttert seinen Hund Bello. (Obama is feeding his dog Bello.)
[Bello][Hund von][Obama] ([Bello][dog of][Obama])

or introduced by a preposition:

e_{23}, e_{26} : Obama, Präsident von Amerika (Obama, president of America)
[Obama][Präsident von][Amerika] ([Obama][president of][America])

The first entity can have an apposition (APP) relation:

e_{11} : Obama, Präsident von Amerika (Obama, president of America)
[Obama][Präsident von][Amerika] ([Obama][president of][America])
e_{13} : Obama, einer der Präsidenten von Amerika (Obama, one of the presidents of America)
[Obama][Präsident von][Amerika] ([Obama][president of][America])

or be the head of the noun kernel:

e_{12} : Obama füttert seinen Hund Bello. (Obama is feeding his dog Bello.)
[Bello][Hund von][Obama] ([Bello][dog of][Obama])

If entity 2 is of type e_{26}, other patterns which are usually for identification for entity 2, may also identify entity 1:

e_{15} : Obamas Mangel an Erfahrung (Obama's lack of experience)
[Obama][Mangel an][Erfahrung] ([Obama][lack of][experience])
e_{16} : sein Mangel an Erfahrung (his lack of experience)
[Obama][Mangel an][Erfahrung] ([Obama][lack of][experience])
e_{17} : dessen Mangel an Erfahrung (whose lack of experience)
[Obama][Mangel an][Erfahrung] ([Obama][lack of][experience])

It has to be mentioned that the attributive possessive pronoun (PPOSAT) in e_{16} and e_{24} cannot be directly used as entity, it refers to real entity which has to be resolved with Co-reference resolution. The attributive relative pronoun (PRELAT) in e_{17} and e_{25} also refers to an entity, but in a relative clause (RC) the referred entity is always the head, so we can simply take the head node of the RC as entity.

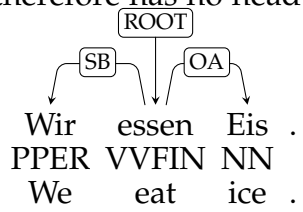
Verb-mediated Fact Extractor

A verb-mediated fact consists of one subject, one verb and an arbitrary number of objects. The dependency grammar offers a relatively easy way to get these components, because the verb is always the root of a phrase in the tree, and the subject is an explicitly labelled child.

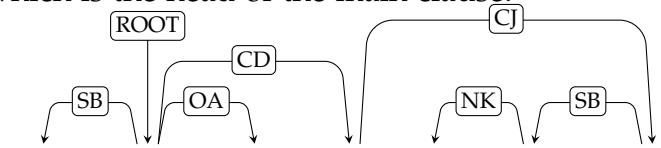
One of our requirements for facts is that they are minimal (as already suggested by Bast and Haussmann (2013)). Our way to achieve this is to iterate the node elements in the tree bottom-up, and separate extracted facts from the tree when they are expandable for the rest of the sentence.

So every time a verb with a subject appears in a sentence, this relation gets extracted along with the inherent objects. Then, we have to decide if it is a stand-alone fact and can get separated from the dependency tree. This depends on the clause type to which the verb belongs to:

Main clause The main clause is always a self-contained fact, and each correct German sentence consists of at least one. The verb of the first main clause is easily identifiable because it is the root of the dependency tree and therefore has no head.

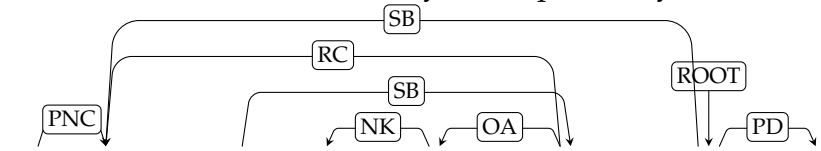


Coordinate clause If there are multiple main clauses joint by coordinating conjunctions, they are called coordinate clauses. Coordinate clauses are of equal status, independent and could stand alone as a sentence. We identify them by checking if the conjunction is not a subordinating conjunction and the same condition is also met for all head phrases. In the following sentence *scheint* is a verb connected by the coordinating conjunction *und* to *essen*, which is the head of the main clause.



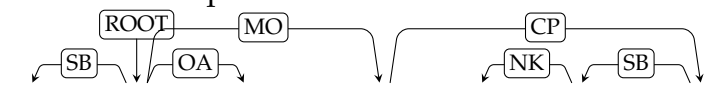
Since all conditions for a coordinate clause are met, *[Sonne][scheint][]* and *[Wir][essen][[Eis]]* can get extracted.

Subordinate clause This clause type depends on a main clause and cannot stand alone in a sentence. It can still contain independent information which we want to extract. A subtype here is the **relative clause**, which is introduced by a relative pronoun. Non-essential (referring to a named entity) relative clauses can always be extracted and separated without loosing any context information. For example *"Bill Gates, der eine Firma besitzt, ist reich."* the fact [Bill Gates][besitzt][Firma] can be extracted from the relative clause. The rest of the sentence *"Bill Gates ist reich."* is still correct and can also be processed. On the other hand, essential relative clauses (referring to a common noun) are crucial for the meaning of the sentence and can not be separated: *"Der Hund, der draußen sitzt, bellt."*. Relative clauses are identified by the Verb-Mediated Fact Extractor by the dependency label RC of the head verb.



As visible

in the sentence above, the subject is the relative pronoun (*der*) which can be replaced by the head of the RC (*Bill Gates*).
Other subordinate clauses are introduced by **subordinating conjunctions** and are never separated from the main clause:



These get also extracted and tagged as *subordinate*, which means that they may be out of context and therefore wrong. In the example above the fact *[Sonne][scheint][]* is valid, although it is from a subordinate clause introduced by *weil*. If subordinating conjunction were *wenn*, the fact would be wrong.

Hentschel and Weydt (2003, pp. 293-305) describe eleven semantic categories of conjunctions:

- Copulative conjunctions (und, sowie, wie...) are all coordinating, thus these will not occur here. They are used to chain words, phrases or sentences together, for example "Wir kaufen Äpfel und Birnen." (We will buy apples and pears.)
- Disjunctive conjunctions (oder, beziehungsweise...), also coordinating, connect multiple possibilities, and often only one is possible: Wir kaufen Äpfel oder Birnen. (We will buy apples or pears.)
- Adversative conjunctions (aber, doch, sondern, wohingegen...) are used to express contradictions. This means that the coordinating phrase is also true when seen in isolation: "Er ist nicht gestern, sondern schon vorgestern angekommen." (He did not arrive today, but already yesterday.)
- Final conjunctions (damit, dass) express a purpose or intention: "Ich schreibe alles auf, damit ich es nicht vergesse." (I write everything down, so I do not forget it.) Phrases introduced by these conjunctions do not contain facts.
- Causal conjunctions (da, weil...) express the reason or cause of the main phrase, which means they also constitute independent facts: "Das Fußballspiel findet in der Halle statt, da es heute regnet." (The soccer game takes place in the soccer hall, since it is raining today.)

- Conditional conjunctions (falls, wenn...) express a condition, which means they are not appropriate for fact extraction: “Ich esse, wenn ich hungrig bin.” (I will eat when I am hungry.)
- Consecutive conjunctions (sodass, ohne dass/zu, um zu ...) state a result or consequence: “Er aß zu viel Schokolade, so/ohne dass ihm schlecht wurde.” (He ate too much chocolate, so that he got sick). A “sodass” phrase can be used, “ohne dass/zu” also but it is negating the meaning, so a “nicht” (not) has to be added for this. The other consecutive conjunctions are meaningless without the main phrase.
- Concessive conjunctions (obgleich, obwohl, trotzdem...) express also reasons but rather in the form of concessions: “Obwohl er krank war, ging er schwimmen.” (Although he was ill, he went swimming.) These phrases are also correct when viewed in isolation.
- Modal conjunctions (indem, anstatt, sofern, als...) denote circumstances, means and manner. As subordinate phrases introduced by these conjunctions often make sense without the main phrase, they should be neglected: “Ich konnte nichts tun, außer die Polizei zu rufen.” (I could not do anything but to call the police.)
- Temporal conjunctions (während, als, nachdem, bevor...) express temporal conditions, here only phrases with conjunctions which denote a current or past event are of interest: “Er war zu Hause, als er starb.” (He was at home when he died.)
- Meaningless conjunctions (dass, ob, wie...) have no special meaning, they are simply used to introduce a subordinate clause: “Ich dachte, dass alles in Ordnung sei.” (I thought everything was alright.)

The Verb-mediated Fact Extractor uses the semantic meanings to decide whether a fact from a subordinate clause should get extracted, most of the adversative, causal, concessive and temporal conjunctions are allowed, the others get skipped. The list of used conjunctions is displayed in table 4.6. We analysed the results in the CoNLL 2009 trial dataset and found that this restrictive approach is very accurate, all of the remaining (subordinate) facts were true, but it reduced the number from 125 to 21. Nearly all of the false negative subordinate clauses had either explicit or implicit “dass” (that) conjunctions. To improve recall, one would have to decide whether facts in “dass” introduced phrases can get extracted. This would be possible by analysing the context (“He knows, that” versus “He believes, that”),

but we leave that for later work. Overall 444 verb-mediated relations were identified in the CoNLL 2009 trial dataset, 338 of these belonged to the main clause of a sentence. The other relations were 22 of coordinate clauses, 21 of subordinate clauses and 63 of relative clauses. 42 of all verb-mediated relations had a named entity as subject, hence were concrete.

Table 4.6.: Conjunctions used by the Verb-Mediated Fact Extractor

Semantic Category	*Instances used by the Verb-Mediated Fact Extractor*
adversative conjunctions	aber, allein, doch, hingegen, jedoch, sondern, während, wohingegen, indes, indessen
causal conjunctions	denn, da, weil, zumal
concessive conjunctions	obgleich, obschon, obwohl, obzwar, trotzdem, wenngleich, wiewohl
temporal conjunctions	als, während, indes, indessen, nachdem, seit, seitdem

4.3.3. Conjunction Processor

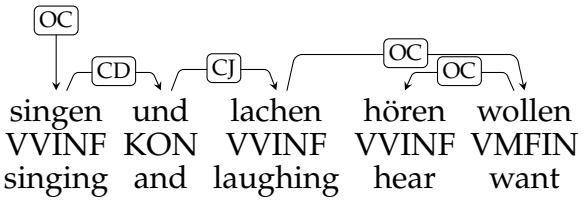
We decided to handle conjunctions after the extraction of facts, to separate responsibilities of our modules. All coordinating conjunctions (und, sowie, wie, aber, doch) except disjunctive once are involved here. We do not process compound conjunctions like “weder ... noch” (either ... or) or “nicht nur ... sondern auch” (not only ... but also), because these occur rather infrequent and are more difficult to handle. The following conjunctions are processed by the Conjunction Processor: und, sowie, aber, allein ,doch, hingegen, jedoch, sondern, denn.

The idea behind this is again to ensure minimality. Facts like [Akim][likes][Pizza and Spaghetti] should get split to [Akim][likes][Pizza] and [Akim][likes][Spaghetti]. The conjunction can occur in any part of the fact, but we only look for conjunctions which are within the first level of the dependency tree of the

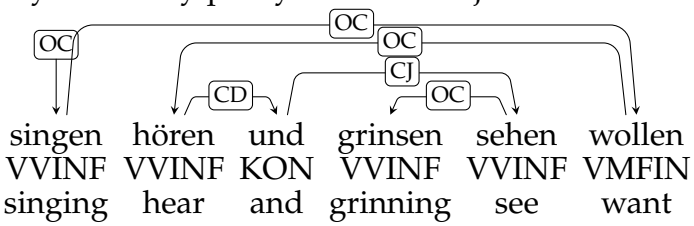
subject, the relation or the object(s). This means that [Akim][likes][green and red apples] would remain the same, because "green and red" is beneath "apples" in the dependency tree. This restriction was made because this task is quite complex and deeper nested conjunctions are also less important.
We had to make two decisions when processing a conjunction:
Can the joint phrases get separated?
Consider the sentences "Akim, brother of singer Bill and musician Kevin." and "Akim, brother of singer and musician Bill." Due to the extractor's design, we have identical dependency nodes in the fact [Akim][brother (of)][singer (...)], but the sub-tree of the object singer is different. Thus, we check if two different named entities are involved, and if yes, we separate the fact to [Akim][brother (of)][singer (Bill)] and [Akim][brother (of)][musician (Kevin)].
Additional instances of conjunctions which should not be considered here are those from subordinate phrases, which appear as object in the fact of the main phrase: [Kevin][knows][that he can swim]. Here, the Conjunction Processor checks whether a subject appears in the phrase because then it should be disregarded.

How to construct the separated phrase?
It is often enough to just duplicate the fact and use one of the joint phrases in every fact: [Akim][likes][apples and bananas] $\rightarrow$ [Akim][likes][apples] and [Akim][likes][bananas] . But in some cases a word can be meant for both conjuncts, as in "fresh apples and bananas". The problem is that even for humans it's sometimes hard to decide how to handle such phrases, and the dependency grammar does not help at all, because it only depicts the relation to its head word. We decided to only handle those which are essential for correct understanding:

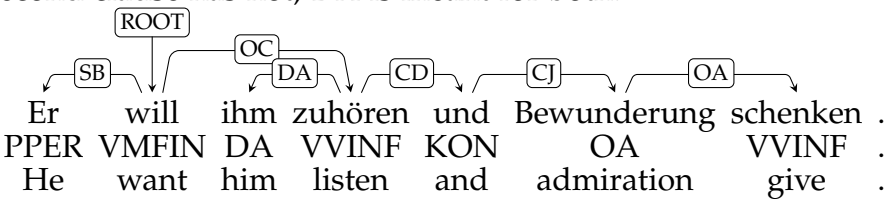
- Genitive attribute: Besitzer und Eigentümer des Kindergartens $\rightarrow$ Besitzer des Kindergartens , Eigentümer des Kindergartens
- Separable verbal particle: tritt und schlägt ein $\rightarrow$ tritt ein, schlägt ein
- Subordinating conjunctions: "because A and B" $\rightarrow$ "because A" and "because B"
- Full, auxiliary and modal verbs: these are often used for both verb conjuncts. In case that all of those verbs are used for both conjunctions, they both appear in the sub-tree of the second verb:



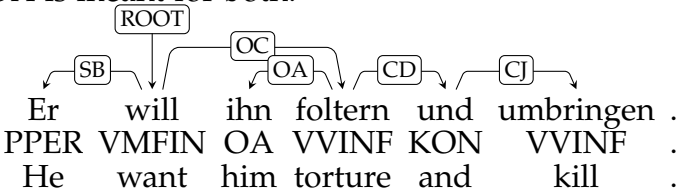
The other possibility is that the full, auxiliary and modal verbs are only used only partly for both conjuncts:



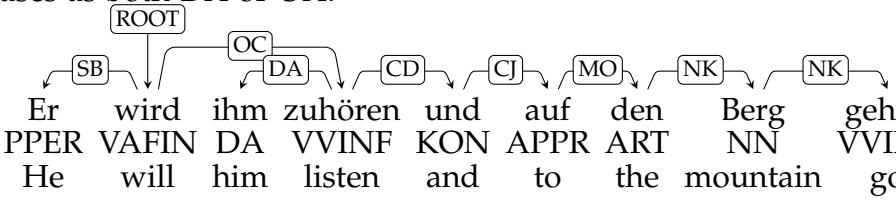
- Accusative and dative: If first clause has a dative object (DA) and second clause has not, DA is meant for both:



If first clause has only accusative object (OA) and the second has not, OA is meant for both:



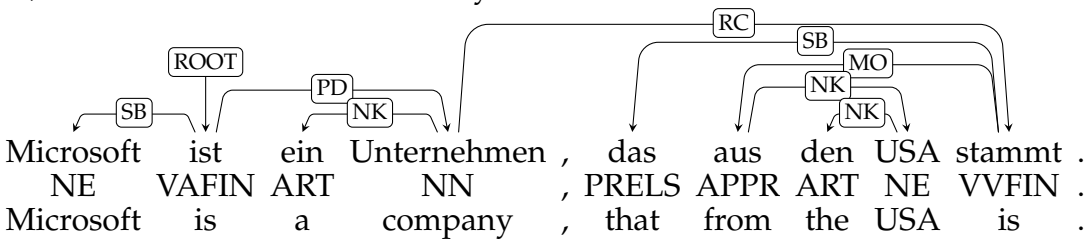
A modifier (MO) with preposition (APPR) counts in the previous two cases as both DA or OA:



When investigating the conjunctions in the CoNLL 2009 trial dataset, we also found that they should not get separated if a preceding "between" exists: "a difference between A and B.", so we added exceptions for words like "between". Overall, 122 conjunctions were correctly processed, but 3 of them did lose important information. Two things were accountable for this. Firstly, there are is relations were the right entity is in singular, which means that this conjunction should be left untouched: "A and B are a team." Our POS tags do distinguish between singular and plural, and also not between different kinds of articles, so additional information would be necessary here to prevent such errors. As this occurred only one time, we did not invest any further effort here. Secondly, it is not always obvious which objects can be used for both conjuncts. Here, we can only process the most frequent and probable word orders. This leads sometimes to incomplete extractions. Another issue was that the number of the verb has to be the same as the number of the subject, so after two conjuncts in the subject were separated and the number changed to singular, the verb would also have to be in singular form. Since the fact is still comprehensible, this was not solved.

4.3.4. Relative Pronoun Processor

The Relative Pronoun Processor is responsible for postprocessing the facts extracted from relative clauses (RCs). As noticeable in the following example, the head of the RC is the entity it refers to:



The pronoun is in this case the subject (SB), but it could also be an object, hence we use the POS tag for identification. Three types of pronouns occur, which have to be handled differently:

- Substituting relative pronoun (PRELS): "der Mann, der in Graz arbeitet" → [in Graz][arbeitet][~~der~~ der Mann]
- Attributive relative pronoun (PRELAT): "der Mann, dessen Freund in Graz arbeitet" → [in Graz][arbeitet][~~dessen Freund~~ der Freund des Mann(s)]
- Adverbial interrogative or relative pronoun (PWAV): "Das Haus, wo er wohnt, ist groß." → [er][wohnt][~~wo~~ im Haus]

4.3.5. Proposition Generator

A proposition generator is used to form facts into the proper output format. Del Corro and Gemulla (2013) described how to generate multiple propositions out of a single fact (by deciding which objects are optional, and creating multiple combinations). We did not tackle this problem, because this was not seen as essential here. Instead, we have a default n-ary generator, which prints the facts as n-ary. This preserves all textual information. In addition, an experimental proposition generator was implemented, which categorises the fact content into *who*, *what*, *where*, *when*, *why*. This was achieved by taking the information of the edge labels, POS tags and the words themselves. For example, if the head of the word has the label *Modifier* and the POS tag *APPR* and the word *in* (in), we assume the phrase belongs to the category *who*.
[wer: Albert Einstein][was: starb][[wann: 1955][wo: in Princeton]]
[who: Albert Einstein][what: died][[when: in 1955][where: in Princeton]]

We discarded our intent to merge multiple object phrases into a single phrase, which would lead to a triple representation. The reason for this was that sometimes the word order in the triple has to be different than it was in the sentence.

5. Evaluation

This chapter describes the evaluation of GerIE. Firstly, we explain how the German datasets were created, and how we annotated the sentences with gold facts. Furthermore, the results of the evaluation are described and discussed.

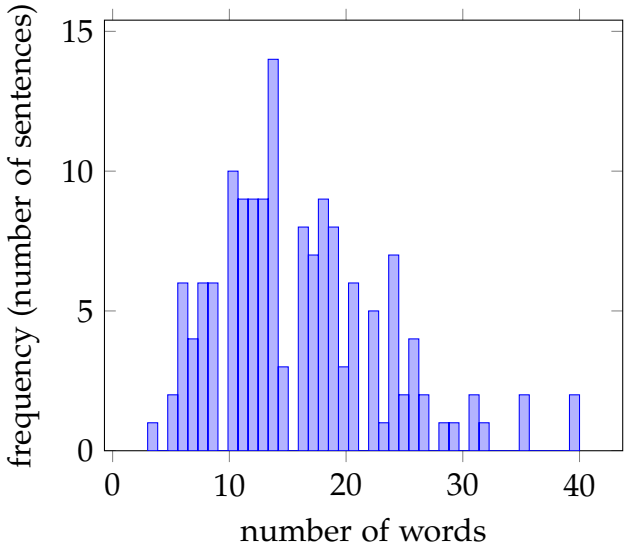
5.1. Datasets

The first dataset was created by randomly selecting 150 sentences from a collection of German news articles. This collection contained 60 articles which were gathered around 2014 from 17 different websites providing news in German language, such as http://www.faz.net/, http://diepresse.com, http://german.ruvr.ru, http://europa.eu/ or www.ukrinform.ua/. The articles consisted on average of 603 words. In this chapter, we will refer to 150 news sentences (including their annotations described in 5.1.1) as *GerNews*.

The sentence length in GerNews ranges from 3 words to 40 words, the average number of words is about 16, a detailed histogram is shown in figure 5.1. This is expectable for the news domain, as the upper limit for a sentence to be easily understandable is approximately 17 words (Groeben and Vorderer, 1982, p. 179). Due to the domain, the dataset contains more direct or reported speech as usual and also includes some not well-formed sentences (which do not satisfy the classical subject-predicate-object structure).

We looked to find out how GerIE performed on the domain of classical printed encyclopaedias as well, so we built the second dataset out of Brock-

Figure 5.1.: Distribution of number of words in the sentences of GerNews

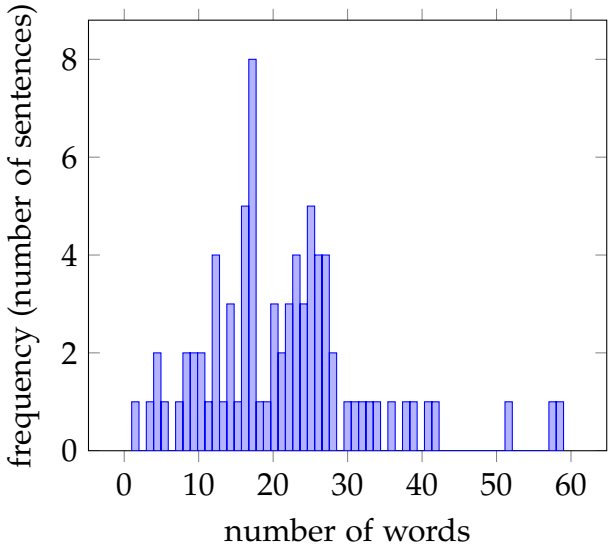


haus [1] articles. Brockhaus is the largest German-language encyclopaedia printed in this century. Its writing style differs from usual text because the goal was to put as much information as possible in preferably little space, because every additional page would increase printing costs. This means that words which are not essential do not occur in the sentences, so they are highly informative. A disadvantage here is that although those sentences are still comprehensible for humans, they are harder to interpret for a dependency parser trained on usual sentences, because, for example, they often lack verbs or pronouns and names are often abbreviated: "Aitinger, Sebastian, Sekretär; geboren in Ulm im September 1508, ..." (Aitinger, Sebastian, secretary, born in Ulm in September 1508, ...).

We randomly selected articles until we had 80 sentences. Note that we skipped articles which were shorter than 10 words, as they often just referred to another article. The length of the sentences ranges from 1 to 59, with an average number of 21.4 words per sentence. Although this dataset contains more sentences which are just headers, hence very short, the average length is still higher compared to GerNews. This indicates a high complexity of

[1] http://www.brockhaus.de/

Figure 5.2.: Distribution of number of words in the sentences of GerBH



the sentences in this dataset. From now on we will refer to this dataset as *GerBH*.

5.1.1. Annotation

The goal was to annotate each sentence with all possible distinct facts which can be extracted from it. It should enable automatic evaluation and calculation of a precision and a recall value. Recent systems (Bast and Haussmann, 2013; Del Corro and Gemulla, 2013) eschewed to use gold facts and just labelled each extraction manually as correct or incorrect, but we were interested in automating the evaluation process as much as possible. Gold facts also make it possible to calculate a recall value.

To make the gold facts as consistent as possible we defined several requirements:

Syntax: We decided to use the form [subject][relation][object][relation$_2$] for each gold fact. Object and relation$_2$ can be empty (e.g. [Einstein][died][][]). Relation$_2$ can contain the second part of a relation,

which can be used either as part of the relation or as part of the object: in the fact [Einstein][likes][in summer][to swim] "to swim" could be either attached after "likes" or before "in summer".

Type: A gold fact has to belong to one of the 5 types GerIE supports (is fact, has fact, preposition fact, noun-mediated fact, verb-mediated fact). Without this restriction, the number of possible gold facts would be unknown, because it is very subjective what parts of a sentence could constitute a fact.

Minimality: To reduce the number of gold facts, we decided that a gold fact has to be minimal. This requirement only affects the main items in subject, relation and object. For example *[He][buys][red and yellow apples]* is minimal, because the main item of the object is "apple", while "red and yellow" are just describing it, so we do not want to separate them. *[He][buys][apples and bananas]* on the other hand, has two main items as objects, so here two gold facts, one for each object, are required. In instances where the fact would lose its meaning or makes no sense if the conjunct phrases were separated, we do not apply this rule: *[John and Tim][are][a team]* is a minimal fact. Additionally, a fact should not occur in another fact, only if the other fact would lose its meaning without it.

Word form: Only words from the sentence are allowed in a gold fact, and they must have exactly the same form. This can lead to grammatically false facts when words have a different case in the sentence and in the fact, for example ("John's car is blue" $\rightarrow$ [John's][has][car], because in German "John's" would be Genitive and written as "Johns") or different number ("John and Tim work in America" $\rightarrow$ [John][work][in America]). This is necessary because GerIE does also not alter any words, and facts will be checked for equality to gold facts. An exception here is the implicit "is" in is facts, "has" in has facts and "of" in noun-mediated facts.

Word selection: It would be possible to write several gold facts each with for example a different combination of adjectives for the noun. It is hard to decide which words are really essential in a fact, so we decided to neglect this task for our gold facts and just use all occurring words which fit in the fact. From the phrase "America's hard-working president Obama..." a gold fact would be *[Obama][hard-working president of][America][]*. This obviates the need to create multiple similar gold

facts and ensures that our facts stay as distinct as possible.

Distinct facts: One way to ensure distinct facts was "word selection" described above. Additionally, it was decided that facts which can be inferred from other facts should not be part of the gold facts. In the example *[Obama][hard-working president of][America][]* the two facts [Obama][is][hard-working president][] and [America][has][hard-working president][] can be inferred, hence they are no gold facts. This always applies when a noun-mediated gold fact exists.

Word order: The order of the words in a gold fact should be equal to their order in the sentence.

Implicit references: We do not include phrases which are only referenced to implicitly in the gold phrases. An example for this is in the sentence "He visited India, talked to president Mukherjee.", where a human identifies that Mukherjee is president of India, but our dependency parser does not treat such things. As we have no possibility to detect such references, we only accept gold facts which can be identified with help of our dependency parser.

We also wanted to know how useful extracted facts are. It is clear that OIE systems can extract large amounts of relations, but we were interested in the usefulness of those extractions. We decided on 4 categories, where it is possible to assign one category to each gold fact in a relatively clear way:

Not Useful: A category for all facts which were seen as non informative for us. This applies to overly specific facts ([Kofi Annan][required][from Goodluck Jonathan the assurance, that they will accept the election outcome][]), too general facts, often because the context is missing ([refugees][live][with rebels][]) or when the relation is too unspecific ([elections][are][in four weeks in Nigeria][]).

Abstract: The category Abstract is for has facts, preposition facts and noun mediated facts. If one of the two objects in the relation is abstract (no real-life object), the fact is seen as abstract, for example [terrorist][has][hate][], [events][in][Mariupol][] or [Moscow][has][notion][]. As observable in those examples, this can also apply when a named entity is involved.

Concrete Named Entity: The most interesting kind of facts because these kinds of facts are usually targeted by traditional IE systems. A concrete fact usually contains specific information about a named entity, such as

[Kofi Annan][is][Nobel peace laureate][] or [Nigeria][has][politicians][], but we also accept references to named entities here: [she][lives][in Berlin][].

General Knowledge: All facts which provide concrete knowledge about things in the world and which are not only valid for a short timespan: [Ukrainians][work][on Russian construction sites][], [UDID][serves][real-time tracking of iPhones][].

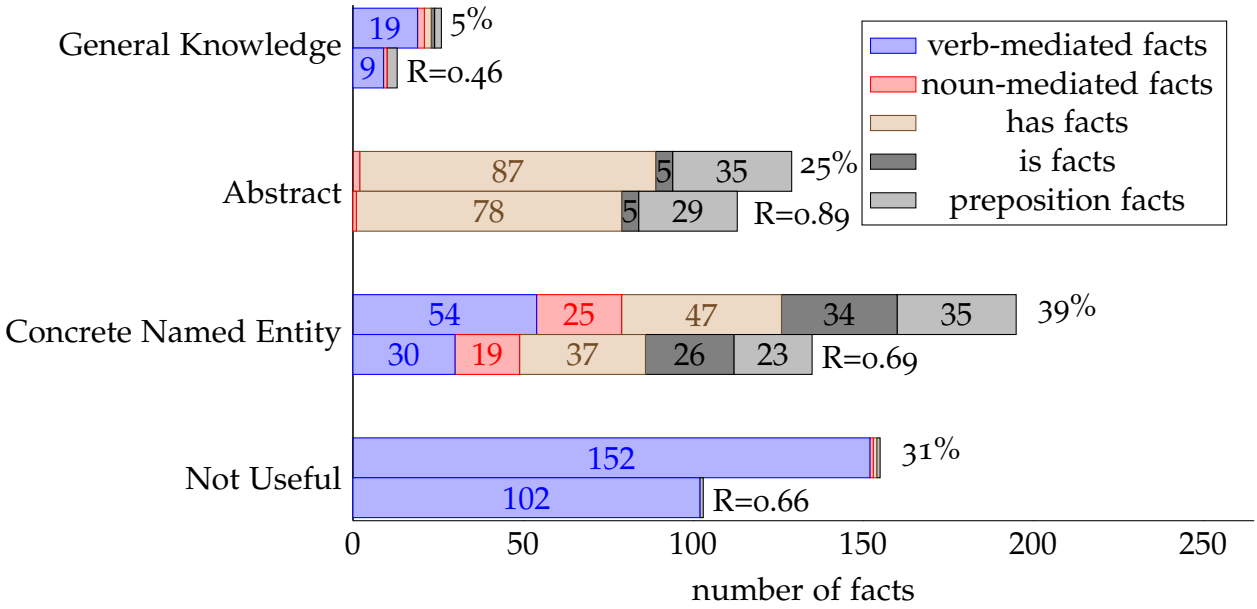
We will use this information in the evaluation to get an overview how well GerIE performs in each category, for example if it achieves significantly higher recall in a specific category.

5.1.2. Gold facts

We annotated 506 gold facts in GerNews. This means every sentence contains on average 3.37 relatively distinct facts. Figure 5.1.2 shows the distribution of facts in GerNews among the 4 categories. 31% of these facts were labelled as Not Useful, 25% as Abstract, 39% were assigned to Concrete Named Entity, and 5% were considered General Knowledge. This means that approximately a third of the correctly extracted facts from news will not be useful for us, which still leaves a high rate of about two useful facts per sentence on average. Not Useful consists nearly only of verb-mediated facts. The reason for this is, that verb-mediated facts which are not concrete are never assigned to Abstract, because they are too specific due to their arbitrary verb as relation. Has facts and preposition-mediated facts, were nearly always assigned to Abstract when they did not contain specific information, because they have a small and fixed set of possible relations.

The distribution of facts in GerNews among different types is displayed in figure 5.1.2. Verb-mediated facts (44%) dominate, because nearly all sentences contain at least one of this type. Has facts (27%) are quite common too, there is about one per sentence on average in both datasets . If preposition facts (15%) are added to has facts (because they are actually a sub category of these), they are about as commonly occurring as verb-mediated facts. Is facts (8%) and noun-mediated facts (6%) occur fewest of all. While preposition facts, is facts, has facts and noun-mediated facts are mainly

Figure 5.3.: Distribution of types among categories in GerNews. Upper bars describe gold facts, lower bars describe correctly extracted facts.



composed of Concrete Named Entity, General Knowledge and Abstract, verb-mediated facts contain Concrete Named Entity, General Knowledge and Not Useful. Preposition facts, has facts and verb-mediated facts have a relatively small part of Concrete Named Entity or General Knowledge facts, approximately half or less. Is facts and noun-mediated facts on the contrary provide mostly useful information, Concrete Named Entity and General Knowledge represent over 87% here.

Although the GerBH dataset is smaller, we could also annotate 452 gold facts there, leading to a large rate of 4.52 facts per sentence. Figure 5.1.2 shows the distribution of categories among types in this dataset. Compared to GerNews, fewer Not Useful facts appear in this dataset (only 9%), while the number of General Knowledge facts is much higher (19%). Concrete Named Entity (44%) and Abstract (28%) are similarly frequent as in GerNews.

In figure 5.1.2 you can see the distribution of categories among types in GerBH. The distribution is similar to the distribution in GerNews: 38% verb-mediated facts, 20% has facts, 18% preposition facts, 16% is facts and

Figure 5.4.: Distribution of categories among types in GerNews. Upper bars describe gold facts, middle bars describe correctly extracted facts and lower bars the incorrect facts (which are not divided into categories).

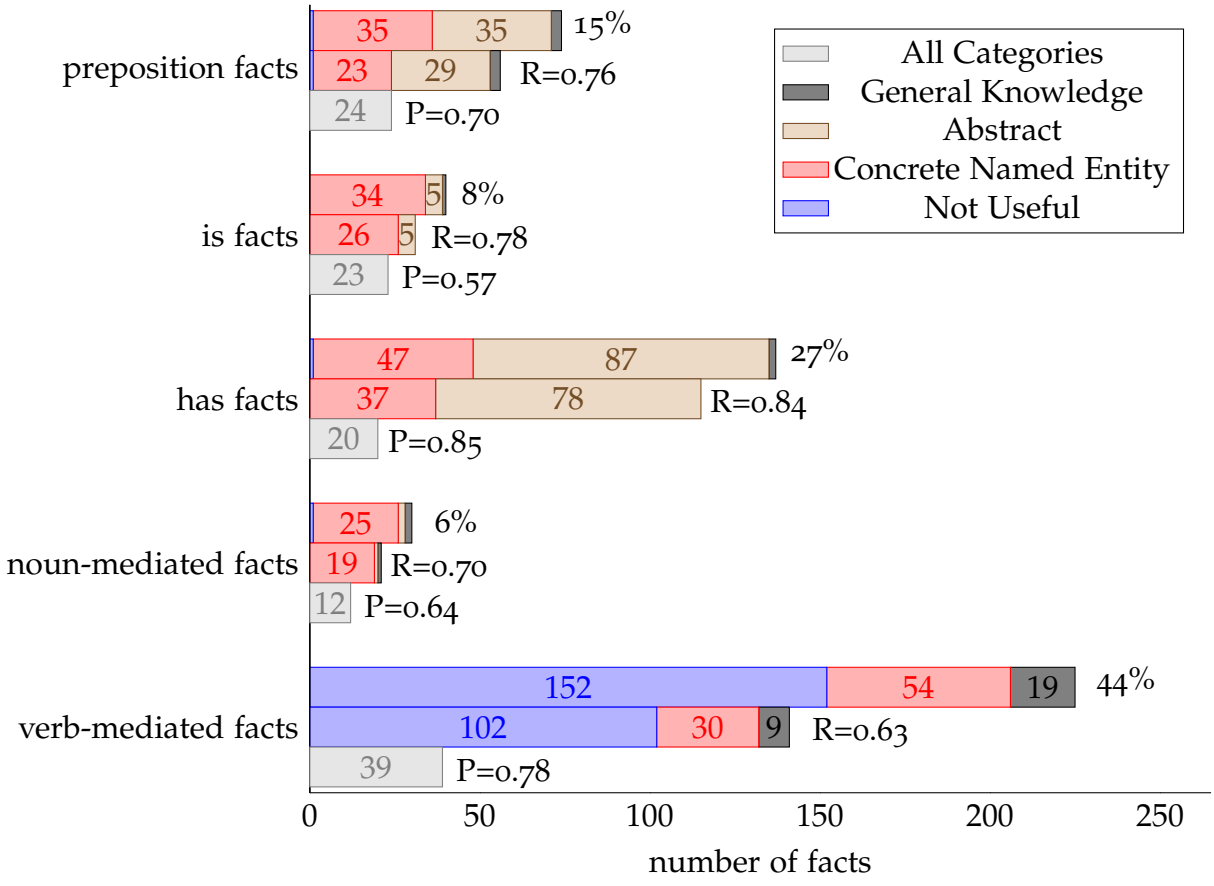
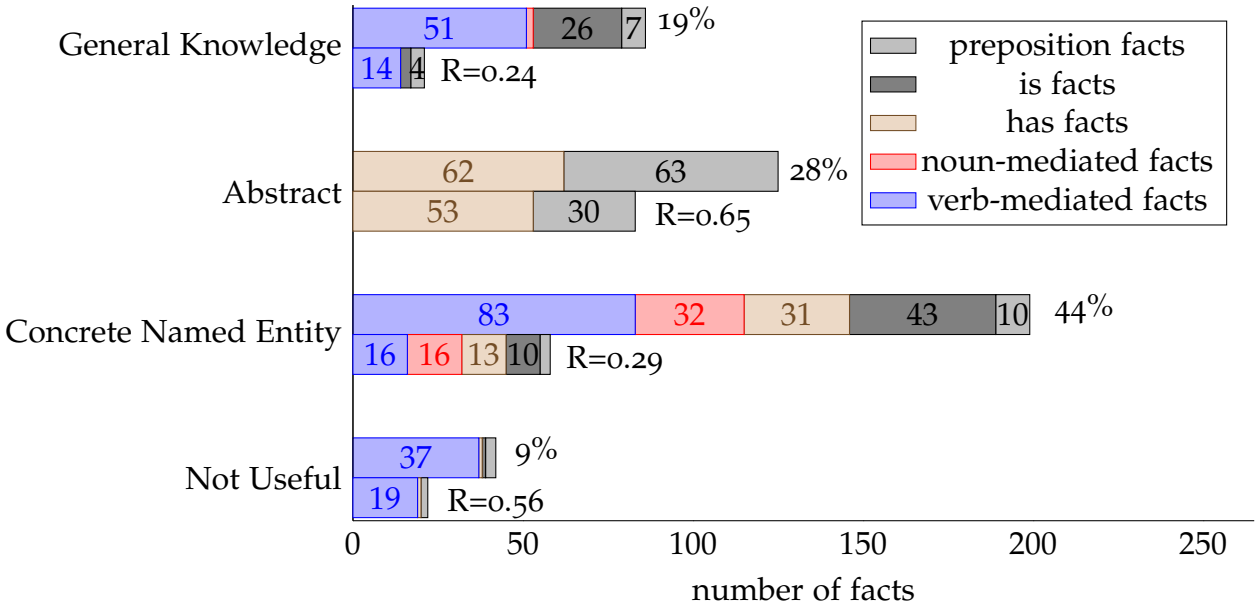


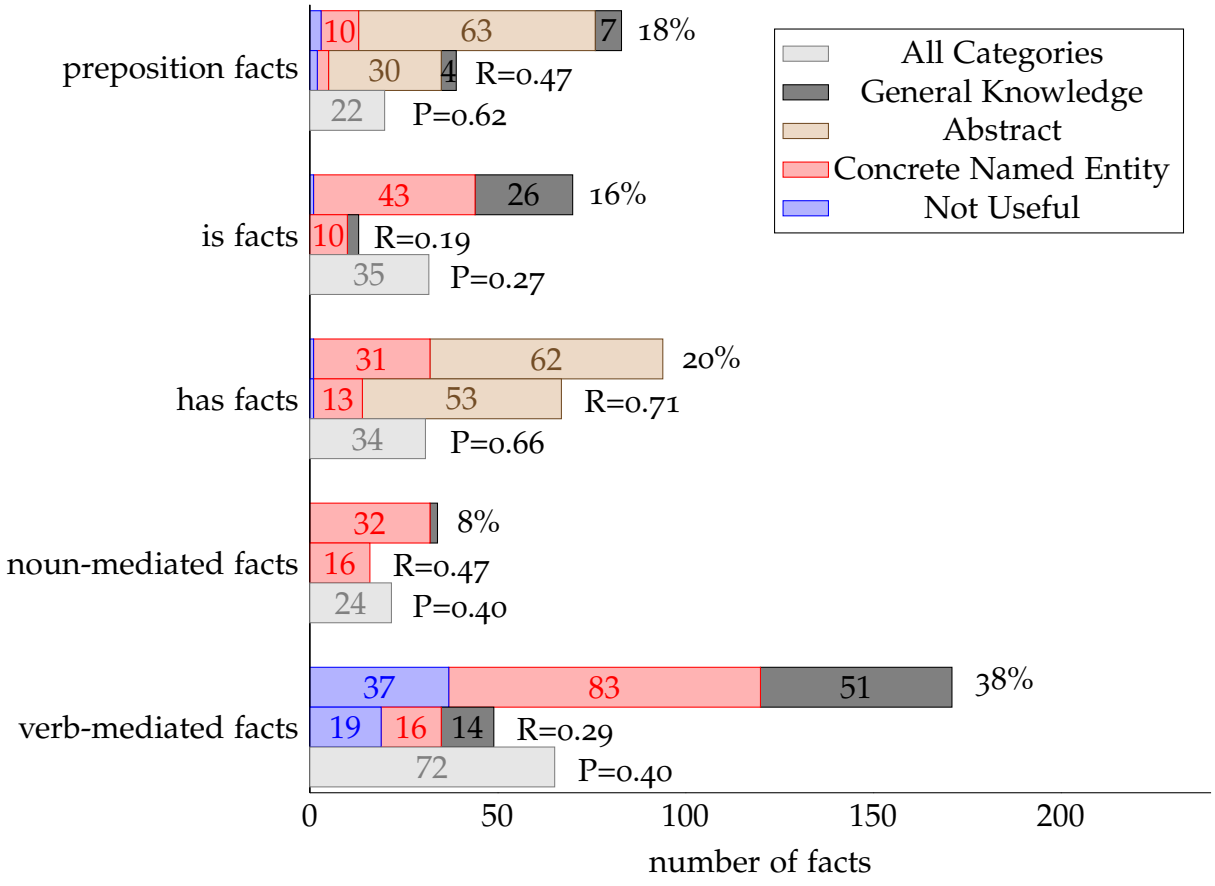
Figure 5.5.: Distribution of types among categories in GerBH. Upper bars describe gold facts, lower bars describe correctly extracted facts.



8% noun-mediated facts. GerBH contains proportionally twice as many is facts as GerNews, which is because of the writing style there. Sentences in this dataset often do not contain verbs when the reader can understand it without it too, such as in "John B., secretary, ..." Here, a usual writer would maybe use "employed as", which would result in a verb-mediated facts([John B.][employed as][secretary]), but in GerBH these are is facts ([John B.][is][secretary]).

During the annotation task, it was found out that some relations are a combination of a verb and a noun. As our verb-mediated facts do not allow nouns in the relation, these relations are split in our gold facts: [He][lends][him his support]. A better way would be to have "lends his support to" as relation, but as the current fact types are specified differently, we separated them in our gold facts.

Figure 5.6.: Distribution of categories among types in GerBH. Upper bars describe gold facts, middle bars describe correctly extracted facts and lower bars the incorrect facts (which are not divided into categories)



5.2. Evaluator

We implemented an evaluator which automatically compares the facts extracted by GerIE to the gold facts. The comparison function is relatively tolerant and ignores minor differences:

- Non word characters are considered irrelevant. This is necessary because, for example, punctuation or quotes may differ, but it does not change the meaning: [He][calls][[her][Mary]] is equal to [He][calls][her "Mary"][]
- Additional white-space characters are ignored.
- Order of the objects is free. GerIE outputs n-arys ([He][gave][[her][a kiss]]) while the gold facts are not ([He][gave][her a kiss][], to make the annotation process easier). That is why the evaluator just checks for each object if it occurs in the gold object, and if size of both facts is the same.

Due to the reason that all gold facts are minimal, extracted facts which are not minimal are labelled as incorrect, although they may actually be correct. This means that all incorrect facts have to be checked manually and labelled as actually incorrect or correct but not minimal. This approach was chosen to exclude gold facts which are just different combinations of similar phrases, which would be very time consuming to annotate, and also render the recall value useless.

5.3. Results

5.3.1. GerNews

Table 5.1 shows that GerIE extracted in total 396 correct facts from the GerNews dataset, resulting in a precision of 0.77 and a recall of 0.78. 14 of the correct facts were not complete, and 16 not minimal. The number of extracted facts per sentence is on average 3.37, which means that more than two correct facts per sentence were obtained. Since GerIE's performance depends on the performance of the dependency parser, we also tagged

Table 5.1.: Results of evaluation

	GerNews	*GerBH*
sentences	150	100
gold facts	506	452
per sentence	3.37	4.52
extracted facts	512	407
per sentence	3.41	4.07
correct, minimal, complete	364	184
correct, minimal	14	9
correct, complete	16	24
incorrect	118	187
precision	0.77	0.54
recall	0.78	0.48
F1	0.77	0.51
gold facts from correctly parsed phrases	446	241
extracted facts from correctly parsed phrases	434	249
precision	0.91	0.88
recall	0.88	0.90
F1	0.89	0.89

each fact with the information whether the underlying phrase was parsed correctly. As a result, we could calculate a precision and recall value for a filtered set of facts which excluded those caused by Mate tools. So when only considering facts from correctly parsed phrases, GerIE achieved 0.91 precision and 0.88 recall.

We were also interested in the comparison of our different fact types and how well they get extracted. Table 5.1.2 shows that verb-mediated facts and noun-mediated facts have the smallest recall value, with 0.63 and 0.70 respectively. The other types all achieved a recall higher than 0.76. These results reflect the values shown in table 5.1.2, which display the distribution of facts among the assigned categories. *Abstract* facts got a high recall (0.89), as they only consist of has facts, is facts, preposition facts and noun mediated facts. Concrete Named Entity facts obtained 0.69 recall, and Not Useful 0.64,

while General Knowledge achieved worst results with a recall of 0.46.

Since incorrect extractions also belong to a specific type, because it is given by the pattern which led to the incorrect extractions, we could also calculate the precision of the individual fact types. Has facts and preposition facts obtained a precision value of 0.66 and 0.64, verb-mediated and noun-mediated facts both 0.4, while is facts had the lowest precision, with 0.27.

Precision distribution among the categories is not known, because incorrect facts would need to be manually assigned to a category after the extraction process, which we did not do.

The error analysis in table 5.2 shows that erroneous parsed sentences were the main reason (66%) for incorrect facts. The second cause (24%) was that some patterns were too unspecific and extracted incongruous facts. This mainly affected has facts and preposition facts. For example, some noun compounds like "Turboprop-Antriebe" (Turboprop Engine) do not comprise a has fact. Often, preposition facts were completely nonsensical when taken out of context, such as *[Nacht][zum][Freitag]* ([night][to][Friday]). Further error sources were that sometimes essential phrases or words were missing (7%), mainly when a conjunction was handled wrongly (*[dispute][between][Putin]*) and that facts were extracted from subordinate clauses which were incorrect without their context (3%). 3% is really low here, which proves that GerIE's list of conjunctions which probably introduce an independent sentence, works well.

Table 5.3 displays the causes for missed facts. The top reason was again the dependency parser (63%). 24% were missed because GerIE thought that they were not independent, based on its list of accepted conjunctions. The rest occurred due to implicit relations ("He wants to be tall like John" $\rightarrow$ *[John][is][tall]*), filtered facts (which were for example in questions) and others like missing patterns. We did, for example, not cover a parenthesis like "Taj Mahal (UNESCO World Heritage Site)", which implies *[Taj Mahal][is][UNESCO World Heritage Site]*.

5.3.2. GerBH

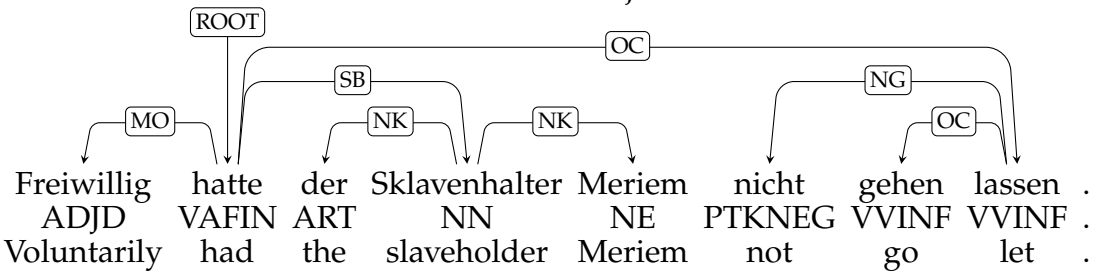
GerIE performed worse with GerBH, it could only extract 217 correct facts (see table 5.1). This means a low recall of 0.48. 9 of the extracted facts were correct but not complete, and 24 correct but not minimal. The total number of extracted facts was 407, leading to 4.07 facts per sentence, and also a low precision value of 0.54. But nearly all of the missed or incorrect facts were caused by incorrectly parsed sentences (see table 5.2 and 5.3). Considering only facts from correctly parsed phrases resulted in 0.88 precision and 0.90 recall.

Table 5.1.2 shows that recall was especially low for is facts (0.17) and verb-mediated facts (0.28). Preposition facts (0.49) and noun-mediated facts (0.42) averaged, while has facts got a surprisingly high recall of 0.75.

The recall of Abstract facts (displayed in table 5.1.2) with 0.65 is also the highest, as those facts consist of has facts and preposition facts only. Not Useful facts achieved 0.53 recall, Concrete Named Entity facts 0.29 and General Knowledge facts 0.24.

The precision among the fact types in GerBH is resembling the distribution in GerNews. Is facts achieved, with a value of 0.27, the lowest precision, verb-mediated and noun-mediated facts reached 0.40, while preposition and has facts obtained the highest values, 0.62 and 0.66.

Incorrect facts (table 5.2) were to 84% caused by erroneously parsed sentences, which cannot be changed by improving GerIE, but rather the model used for Mate tools. An example for this is the following sentence, where "Meriem" was not identified as accusative object:



This led to the incorrect facts *[Meriem][ist][Sklavenhalter]* and *[Meriem][hatte] [Freiwillig][nicht gehen lassen]* and the missed fact *[der Sklavenhalter][hatte]*

[Meriem Freiweilig][nicht gehen lassen]. 4% were found to be incorrect due to too unspecific patterns, such as the extraction of facts from compound words. *[Turboprop][hat][Maschinen]* was for examples incorrectly extracted from "Turboprop-Maschinen". Missing phrases accounted for another 4%, an example for this is the phrase "Uno-Hochkommissariats für Menschenrechte (OHCHR)", where "Uno" misses in the in the extracted fact *[OHCHR][ist][Hochkommissariats für Menschenrechte* (Office of the High Commissioner for Human Rights). 7% originated from subordinate clauses which were incorrect without context: *[jeder][kann][Konsensverfassung][zustimmen]* is only correct within its context in the sentence "Die Oppositionsparteien hingegen wollten eine Konsensverfassung ausarbeiten, der jeder zustimmen kann." Also, nearly all of the missed facts (94%) could not get extracted because of dependency parsing errors, an example was already provided above for incorrect extractions. Other reasons for missed facts were, that subordinate clauses which did contain correct information were rejected because the conjunction indicated that it is probably incorrect ("He said, that ..."). *Implicit* means that the fact could not get extracted because for example the relation was not explicitly written: "Aleppo, Syrien" →*[Aleppo][in][Syrien]*. Some sentences with valid facts were filtered because of their punctuation mark, only full stops were accepted. For example the sentence "Doch wie schützt man Flüchtlinge, die mit den Rebellen leben?" was ignored, although it contains the fact *[Flüchtlinge][leben][mit den Rebellen]*.

Table 5.2.: Error sources of incorrect facts

	GerNews	GerBH
incorrect facts	118	187
erroneous dependency parse	78	158
pattern too general	28	8
essential phrase missed	8	8
subordinate clause not independent	3	12
others	1	1

Table 5.3.: Error sources of missed facts

	GerNews	GerBH
missed facts	95	224
erroneous dependency parse	60	211
subordinate clause rejected	23	1
implicit	3	1
filtered	2	2
others	7	9

5.4. Discussion

The precision and recall of both datasets is quite different, but that was expected due to the different domains. GerNews achieved much better results because the parser (Mate tools) was trained on the Tiger corpus, which also consists of news articles from the Frankfurter Rundschau. Additionally, the sentences in GerBH are quite special and different from those you usually encounter on the Web. This explains why GerIE got 15% of the incorrect and 12% of the missed extractions due to erroneously parsed phrases for GerNews, and more than twice as much in GerBH, 39% of the incorrect and 47% of the missed extractions. The second recall and precision values, based only on facts from correctly parsed phrases, show that GerIE obtains similar values (around 0.89) in both datasets, which shows that the used patterns are domain independent. Note that only a small amount of facts was correct but not minimal or not complete. One reason why there are so few non minimal facts is that our definition of minimal is not very strict, we did not want to handle possible combinations of objects, hence we only checked if an extracted fact may be split into two distinct facts. If there were multiple objects, we always used all of them: *[Einstein][died][[painfully][in Princeton][in 1955]]*. The advantage of this is that the number of gold facts which we had to annotate was smaller and unambiguous. This also caused the low number of incomplete facts: since GerIE usually uses all objects, its more difficult to miss an essential phrase. The comparison of recall and precision between types and categories revealed that has facts achieve above average values in both datasets. This shows that the implemented patterns for has facts work

well in diverse domains, and the simplicity makes them less prone to errors caused by the parser. The "Abstract" category also achieves above average recall, as it mainly consists of has facts. The error sources (table 5.2 and 5.3) revealed several areas where GerIE can be improved. Missing patterns can be easily added, it just is not possible to think of every possibility in advance. We found existing patterns with the help of the Tiger corpus, which explains the fact that GerNews (that is from the same domain) has a very low number of facts caused by missing patterns. At the same time, it is proportionally much higher in GerBH, which has a different domain (hence often differently constructed sentences). Facts which were unnecessarily filtered out because they occurred e.g. in an interrogative clause ("Why did John kill Bill?" → *[John][did kill][Bill]*), are currently not distinguished from those which should get filtered ("Can humans eat rocks?" → *[humans][can eat][rocks]*). Here, it may be possible to add a more complex decision logic and not to simply remove interrogative clauses. For subordinate clauses, for example, we defined a set of conjunctions (see subsection 4.3.2) which have a high probability to introduce an independent subordinate clause. As noticeable in the table, this works very well for GerNews, but GerBH requires a stricter set, there a large portion (when ignoring dependency parse errors) of the incorrect facts were those from subordinate clauses wrongly seen as independent. An example for this problem is the subordinate conjunction "that". In the sentence "John knows that bees gather pollen." a fact can also get extracted from the subordinate clause: [bees][gather][pollen]. If it reads "John thinks that..." instead, the subordinate clause should not get extracted. Patterns which are too unspecific can be analysed and maybe additional constraints added. We found, for example, that preposition facts tend to be incorrect sometimes, (we also marked senseless facts as incorrect) when taken out of context, hence the context has to be considered during extraction, too.

Compared to the two state-of-the-art English OIE systems ClausIE and CSD-IE, the number of extracted facts and the precision are on a similar level, taken the results of GerIE's GerNews evaluation and those published by Bast and Haussmann (2013). All systems extract on average more than 3 facts per sentence. This rate is much higher than it was with previous systems, the main reason for this is that older systems only focused on verb-mediated relations. ClausIE and CSD-IE also extract noun-mediated facts,

is facts and has facts. GerIE additionally covers preposition facts, which composed 11% of all extracted facts. ClausIE and CSD-IE on the other hand, have the advantage that they generate multiple facts instead of one when multiple objects occur. In the previous example, *[Einstein][died][[painfully][in Princeton][in 1955]]*, GerIE sees one single fact, whereas the English systems separate these objects: *[Einstein][died][painfully]*, *[Einstein][died][in Princeton]* and *[Einstein][died][in 1955]*.

6. Conclusion

In this thesis, we ascertained an appropriate approach and architecture for a German OIE system, and compared resources necessary for preprocessing of German texts. To accomplish this, we surveyed existing approaches for OIE and analysed their performance and applicability to German texts. We showed that dependency parser based systems, such as ClausIE, yield the most promising results for German OIE systems, as they enable higher precision and recall than shallow feature based OIE systems. Furthermore, we described how a small collection of rules can easily compete with trained systems. In our comparison of available dependency parsers for German, we chose Mate Tools because of its state-of-the-art performance. We presented GerIE, the first German OIE system, which was implemented based on the approach previously found to provide optimal results. We distinguished between five different types of relations: is, has, preposition, noun-mediated and verb-mediated relations, which GerIE can extract. Preprocessing modules were explained. They are necessary in order to complement the information provided by the dependency parser with additions required in some of the extraction patterns. A total of twelve basic patterns which indicate relations in sentences were found in the dependency structure. We explained the necessary steps to ensure minimal facts. We created two first German OIE evaluation datasets, which showed that GerIE achieves at least 0.88 precision and recall with correctly parsed sentences, while errors made by the used dependency parser can reduce precision to 0.54 and recall to 0.48. The gold facts were assigned to one of four different categories describing their usefulness, which revealed that about half of the facts were concrete, while the other half were either considered abstract or not useful. The results point out that OIE in German is not more difficult than in English. The presented system achieved state-of-the-art results with only a small set of patterns. Dependency parsing provides excellent information for OIE, as it also handles the relatively free word order in German language very

well. The assignment of facts to categories indicated that about half of them are concrete, thus similar to relations targeted by traditional IE systems. GerIE is an important contribution to current IE research, as it enables Open Information Extraction from German corpora which was not possible before. We hope that this will boost further research in this area, as German is a major language in Europe with many available sources (German news, Wikipedia...), which can be processed from un- or semi-structured to structured text this way. Potential future research includes coverage of more patterns, as more uncommon ones not occurring in our datasets are not covered yet. Another interesting future direction would be to distinguish between optional and compulsory objects, and to infer from existing facts to new facts. The number of different relations which are currently extracted is quite large. Here, synonym detection could aid to reduce the set of distinct relations. GerIE does not utilise Named Entity Recognition and Co-reference Resolution yet. For real world application, these tasks should be performed before extracting relations, as they improve usability of the extractions.

Appendix

Appendix A.

Gazetteer Lists

A.1. Negation Words

Based on http://www.canoo.net/services/OnlineGrammar/Satz/Negation/Negationswort/index.html (28.06.2016):

kein keiner nein nicht nichts nie niemals niemand nirgends nirgendwo nirgendwoher nirgendwohin keinesfalls keineswegs mitnichten

A.2. Quantities and Units

Based on https://de.wikipedia.org/wiki/Liste_physikalischer_Gr\unhbox\voidb@x\bgroup\accent127o\penalty\@M\hskip\z@skip\egroup\OT1\ssen and http://www.canoo.net/services/GermanSpelling/Regeln/Gross-klein/Zahlen.html (28.06.2016):

Hundert Tausend Million Milliarde Billion Billiarde Halb Drittel Viertel Fünftel Sechstel Siebtel Achtel Neuntel Zehntel Dutzend Handvoll Paar Meter Jahr Kilometer Gramm Kilogramm Zentner Euro Dollar Pfund Sekunde Stunde Quadratmeter Kubikmeter Hertz Newton Pascal Joule Watt Kelvin Ampere Coulomb Volt Ohm Farad Tesla Henry Candela Gray Becquerel Mol Radiant Siemens Lumen

Bibliography

Agichtein, Eugene and Luis Gravano (2000). "Snowball: Extracting Relations from Large Plain-text Collections". In: *Proceedings of the 5th ACM Conference on Digital Libraries*. San Antonio, Texas, USA, pp. 85–94 (cit. on p. 4).

Akbik, Alan and Jügen Broß (2009). "Wanderlust: Extracting Semantic Relations from Natural Language Text Using Dependency Grammar Patterns". In: *Proceedings of the 2009 Semantic Search Workshop at the 18th International World Wide Web Conference*. Madrid, Spain, pp. 6–15 (cit. on pp. 7, 13, 15, 19, 30, 41, 42).

Akbik, Alan and Alexander Löser (2012). "KRAKEN: N-ary Facts in Open Information Extraction". In: *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. Association for Computational Linguistics. Montreal, Canada, pp. 52–56 (cit. on pp. 8, 13, 20, 29, 30, 40).

Attardi, Giuseppe (2006). "Experiments with a Multilanguage Non-Projective Dependency Parser". In: *Proceedings of the 10th Conference on Computational Natural Language Learning*. New York City, USA, pp. 166–170 (cit. on p. 34).

Banko, Michele (2009). "Open Information Extraction for the Web". PhD thesis. Seattle, Washington, USA: University of Washington (cit. on p. 14).

Banko, Michele, Michael J Cafarella, et al. (2007). "Open Information Extraction for the Web". In: *Proceedings of the International Joint Conference on Artificial Intelligence*. Hyderabad, India, pp. 2670–2676 (cit. on pp. 1, 6, 7, 13, 15, 18, 23, 40, 41).

Banko, Michele and Oren Etzioni (2007). "Strategies for Lifelong Knowledge Extraction from the Web". In: *Proceedings of the 4th International Conference on Knowledge Capture*. Whistler, BC, Canada, pp. 95–102 (cit. on p. 11).

Banko, Michele, Oren Etzioni, and Turing Center (2008). "The Tradeoffs Between Open and Traditional Relation Extraction." In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Columbus, OH, USA, pp. 28–36 (cit. on p. 10).

Bast, Hannah, Florian Bäurle, et al. (2012). "Broccoli: Semantic Full-Text Search at your Fingertips". In: *Computing Research Repository (CoRR)* abs/1207.2615 (cit. on p. 10).

Bast, Hannah and Elmar Haussmann (2013). "Open Information Extraction via Contextual Sentence Decomposition". In: *Proceedings of the 2013 IEEE 7th International Conference on Semantic Computing*. Irvine, CA, USA, pp. 154–159 (cit. on pp. 6, 9, 13, 21, 29, 31, 40, 57, 68, 82).

Björkelund, Anders et al. (2010). "A High-Performance Syntactic and Semantic Dependency Parser". In: *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*. Beijing, China, pp. 33–36 (cit. on p. 38).

Bohnet, Bernd (2009). "Efficient Parsing of Syntactic and Semantic Dependency Structures". In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*. Boulder, Colorado, pp. 67–72 (cit. on p. 36).

— (2010). "Very High Accuracy and Fast Dependency Parsing is Not a Contradiction". In: *Proceedings of the 23rd International Conference on Computational Linguistics*. Beijing, China, pp. 89–97 (cit. on pp. 36, 37).

Bohnet, Bernd and Jonas Kuhn (2012). "The Best of Both Worlds – A Graph-based Completion Model for Transition-based Parsers". In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Avignon, France, pp. 77–87 (cit. on p. 37).

Bohnet, Bernd and Joakim Nivre (2012). "A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea, pp. 1455–1465 (cit. on p. 37).

Bohnet, Bernd, Joakim Nivre, et al. (2013). "Joint Morphological and Syntactic Analysis for Richly Inflected Languages". In: *Transactions of the Association for Computational Linguistics* 1, pp. 415–428 (cit. on p. 37).

Brill, Eric (2003). "Computational Linguistics and Intelligent Text Processing: 4th International Conference, CICLing 2003 Mexico City, Mexico,

February 16–22, 2003 Proceedings". In: Berlin, Germany: Springer Berlin Heidelberg. Chap. Processing Natural Language without Natural Language Processing, pp. 360–369 (cit. on p. 14).

Brin, Sergey (1999). "Extracting Patterns and Relations from the World Wide Web". In: *Selected Papers from the International Workshop on The World Wide Web and Databases*. Valencia, Spain, pp. 172–183 (cit. on p. 4).

Bronzi, Mirko et al. (2012). "Automatic Evaluation of Relation Extraction Systems on Large-scale". In: *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. Montreal, Canada, pp. 19–24 (cit. on p. 42).

Buchholz, Sabine and Erwin Marsi (2006). "CoNLL-X Shared Task on Multilingual Dependency Parsing". In: *Proceedings of the 10th Conference on Computational Natural Language Learning*. New York City, New York, pp. 149–164 (cit. on p. 38).

Carlson, Andrew et al. (2010). "Toward an Architecture for Never-Ending Language Learning". In: *Proceedings of 24th AAAI Conference on Artificial Intelligence*. Atlanta, Georgia, USA, pp. 1306–1313 (cit. on p. 11).

Castella Xavier, Clarissa et al. (2013). "Open Information Extraction Based on Lexical-Syntactic Patterns". In: *Proceedings of the Brazilian Conference on Intelligent Systems*. Fortaleza, CE, Brazil, pp. 189–194 (cit. on pp. 8, 13, 19, 29, 30, 40).

Chinchor, Nancy, David D. Lewis, and Lynette Hirschman (1993). "Evaluating Message Understanding Systems: An Analysis of the Third Message Understanding Conference (MUC-3)". In: *Computational Linguistics* 19.3, pp. 409–449 (cit. on p. 4).

Choi, Jinho D and Andrew McCallum (2013). "Transition-based Dependency Parsing with Selectional Branching". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria, pp. 1052–1062 (cit. on p. 36).

Choi, Jinho D, Joel Tetreault, and Amanda Stent (2015). "It Depends: Dependency Parser Comparison Using A Web-based Evaluation Tool". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Beijing, China, pp. 387–396 (cit. on pp. 14, 36).

Christensen, Janara, Stephen Soderland, Oren Etzioni, et al. (2010). "Semantic Role Labeling for Open Information Extracti". In: *Proceedings of the NAACL HLT 2010 1st International Workshop on Formalisms and*

Methodology for Learning by Reading. Los Angeles, CA, USA, pp. 52–60 (cit. on pp. 8, 11, 13, 29, 32).

Christensen, Janara, Stephen Soderland, Oren Etzioni, et al. (2011). "An Analysis of Open Information Extraction based on Semantic Role Labeling". In: *Proceedings of the 6th International Conference on Knowledge Capture*. Banff, AB, Canada, pp. 113–120 (cit. on p. 17).

Cimiano, Philipp and Johanna Wenderoth (2005). "Automatically Learning Qualia Structures from the Web". In: *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*. Ann Arbor, Michigan, USA, pp. 28–37 (cit. on pp. 9, 13, 20).

Craven, Mark et al. (2000). "Learning to Construct Knowledge Bases from the World Wide Web". In: *Artificial Intelligence* 118.1–2, pp. 69–113 (cit. on p. 4).

De Marneffe, Marie-Catherine et al. (2014). "Universal Stanford Dependencies: A cross-linguistic typology". In: *Proceedings of the 9th International Conference on Language Resources and Evaluation*. Reykjavik, Iceland, pp. 4585–4592 (cit. on p. 28).

Del Corro, Luciano and Rainer Gemulla (2013). "ClausIE: Clause-Based Open Information Extraction". In: *Proceedings of the 22nd International World Wide Web Conference*. Rio de Janeiro, Brazil, pp. 355–366 (cit. on pp. 6, 8, 13, 22, 29–31, 40, 65, 68).

Etzioni, Oren, Michele Banko, et al. (2008). "Open Information Extraction from the Web". In: *Communications of the ACM* 51.12, pp. 68–74 (cit. on p. 44).

Etzioni, Oren, Michael Cafarella, et al. (2005). "Unsupervised Named-entity Extraction from the Web: An Experimental Study". In: *Artificial Intelligence* 165.1, pp. 91–134 (cit. on p. 4).

Etzioni, Oren, Anthony Fader, et al. (2011). "Open Information Extraction: The Second Generation". In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. Barcelona, Spain, pp. 3–10 (cit. on p. 18).

Fader, Anthony, Stephen Soderland, and Oren Etzioni (2011). "Identifying Relations for Open Information Extraction". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, pp. 1535–1545 (cit. on pp. 8, 13, 23, 29–31, 41).

Foth, Kilian A, Michael Daum, and Wolfgang Menzel (2004). *Constraint Solving and Language Processing: First International Workshop, CSLP 2004,*

Roskilde, Denmark, September 1-3, 2004, Revised Selected and Invited Papers. Berlin, Germany: Springer Berlin Heidelberg. Chap. Parsing Unrestricted German Text with Defeasible Constraints, pp. 140–157 (cit. on p. 39).

Gamallo Otero, Pablo and Isaac González López (2011). "A grammatical formalism based on patterns of Part of Speech tags". In: *International Journal of Corpus Linguistics* 16.1, pp. 45–71 (cit. on p. 32).

Gamallo, Pablo (2015). "Dependency Parsing with Compression Rules". In: *Proceedings of the 14th International Conference on Parsing Technologies*. Nara, Japan, pp. 107–117 (cit. on p. 27).

Gamallo, Pablo and Marcos Garcia (2015). "Progress in Artificial Intelligence: 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings". In: Cham, Switzerland: Springer International Publishing. Chap. Multilingual Open Information Extraction, pp. 711–722 (cit. on pp. 6, 9, 13, 22, 32, 40).

Gamallo, Pablo, Marcos Garcia, and Santiago Fernández-Lanza (2012). "Dependency-Based Open Information Extraction". In: *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*. Avignon, France, pp. 10–18 (cit. on pp. 9, 13, 20–22, 32).

Groeben, Norbert and Peter Vorderer (1982). *Leserpsychologie: Textverständnis-Textverständlichkeit*. Stroudsburg, PA, USA: Aschendorff Münster (cit. on p. 66).

Hall, Johan and Joakim Nivre (2008). "A Dependency-Driven Parser for German Dependency and Constituency Representations". In: *Proceedings of the Workshop on Parsing German*. Columbus, Ohio, pp. 47–54 (cit. on pp. 34, 38).

Hentschel, Elke and Harald Weydt (2003). *Handbuch der deutschen Grammatik*. 3rd ed. Berlin, Germany: Walter de Gruyter (cit. on pp. 25, 59).

Horsmann, Tobias, Nicolai Erbs, and Torsten Zesch (2015). "Fast or Accurate?– A Comparative Evaluation of PoS Tagging Models". In: *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology*. Duisburg, Germany, pp. 22–30 (cit. on p. 14).

Johansson, Richard and Pierre Nugues (2006). "Investigating Multilingual Dependency Parsing". In: *Proceedings of the 10th Conference on Computational Natural Language Learning*. New York City, USA, pp. 206–210 (cit. on p. 37).

Kim, Jun-Tae and D.I. Moldovan (1993). "Acquisition of Semantic Patterns for Information Extraction from Corpora". In: *Proceedings of the 9th*

Conference on Artificial Intelligence for Applications. Orlando, Florida, USA, pp. 171–176 (cit. on p. 4).

Kok, Stanley and Pedro Domingos (2005). "Learning the Structure of Markov Logic Networks". In: *Proceedings of the 22nd International Conference on Machine Learning*. Bonn, Germany, pp. 441–448 (cit. on p. 8).

Kübler, Sandra, Erhard W. Hinrichs, and Wolfgang Maier (2006). "Is It Really That Difficult to Parse German?" In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia, pp. 111–119 (cit. on p. 33).

Kübler, Sandra, Ryan McDonald, and Joakim Nivre (2009). "Dependency parsing". In: *Synthesis Lectures on Human Language Technologies* 1.1, pp. 1–127 (cit. on pp. 34, 35).

Kübler, Sandra and Jelena Prokic (2006). "Why is German dependency parsing more reliable than constituent parsing?" In: *Proceedings of the 5th International Workshop on Treebanks and Linguistic Theories*. Prague, Czech Republic, pp. 7–18 (cit. on p. 33).

Lavelli, Alberto (2014). "Comparing State-of-the-art Dependency Parsers for the EVALITA 2014 Dependency Parsing Task". In: *Proceedings of the 4th International Workshop EVALITA 2014*. Pisa, Italy (cit. on p. 36).

Marcus, Mitchell P, Mary Ann Marcinkiewicz, and Beatrice Santorini (1993). "Building a Large Annotated Corpus of English: The Penn Treebank". In: *Computational Linguistics* 19.2, pp. 313–330 (cit. on p. 26).

Mintz, Mike et al. (2009). "Distant supervision for relation extraction without labeled data". In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*. Suntec, Singapore, pp. 1003–1011 (cit. on p. 18).

Moschitti, Alessandro (2006). "Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings". In: Berlin, Germany: Springer Berlin Heidelberg. Chap. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees, pp. 318–329 (cit. on p. 19).

Nakashole, Ndapandula, Gerhard Weikum, and Fabian Suchanek (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. Jeju Island, Korea, pp. 1135–1145 (cit. on pp. 8, 11, 13, 19).

Nilsson, Jens, Sebastian Riedel, and Deniz Yuret (2007). "The CoNLL 2007 Shared Task on Dependency Parsing". In: *Proceedings of the CoNLL shared task session of EMNLP-CoNLL*. Prague, Czech Republic, pp. 915–932 (cit. on p. 38).

Nivre, Joakim (2003). "An Efficient Algorithm for Projective Dependency Parsing". In: *Proceedings of the 8th International Workshop on Parsing Technologies*. Nancy, France, pp. 149–160 (cit. on p. 34).

Oepen, Stephan et al. (2014). "SemEval 2014 Task 8: Broad-Coverage Semantic Dependency Parsing". In: *Proceedings of the 8th International Workshop on Semantic Evaluation*. Dublin, Ireland, pp. 63–72 (cit. on p. 35).

Pasternack, Jeff and Dan Roth (2010). "Knowing What to Believe (when You Already Know Something)". In: *Proceedings of the 23rd International Conference on Computational Linguistics*. Beijing, China, pp. 877–885 (cit. on p. 10).

Petrov, Slav, Dipanjan Das, and Ryan McDonald (2011). "A Universal Part-of-Speech Tagset". In: *arXiv preprint arXiv:1104.2086* (cit. on p. 25).

Petrov, Slav and Ryan McDonald (2012). "Overview of the 2012 Shared Task on Parsing the Web". In: *Notes of the 1st Workshop on Syntactic Analysis of Non-Canonical Language*. Vol. 59. Montreal, Canada (cit. on p. 35).

Piskorski, Jakub and Roman Yangarber (2013). "Multi-source, Multilingual Information Extraction and Summarization". In: Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. Information Extraction: Past, Present and Future, pp. 23–49 (cit. on p. 3).

Pustejovsky, James (1991). "The Generative Lexicon". In: *Computational Linguistics* 17.4, pp. 409–441 (cit. on p. 19).

Rasooli, Mohammad Sadegh and Joel R. Tetreault (2015). "Yara Parser: A Fast and Accurate Dependency Parser". In: *Computing Research Repository (CoRR)* abs/1503.06733 (cit. on p. 36).

Ravichandran, Deepak and Eduard Hovy (2002). "Learning Surface Text Patterns for a Question Answering System". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 41–47 (cit. on p. 4).

Richardson, Matthew and Pedro Domingos (2006). "Markov logic networks". In: *Machine Learning* 62.1-2, pp. 107–136 (cit. on p. 18).

Riloff, Ellen (1996). "Automatically Generating Extraction Patterns from Untagged Text". In: *Proceedings of the 13th National Conference on Artificial Intelligence*. Portland, Oregon, pp. 1044–1049 (cit. on p. 4).

Riloff, Ellen and Rosie Jones (1999). "Learning Dictionaries for Information Extraction by Multi-level Bootstrapping". In: *Proceedings of the 16th National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*. Orlando, Florida, USA, pp. 474–479 (cit. on p. 4).

Rosenfeld, Benjamin and Ronen Feldman (2006). "URES: An Unsupervised Web Relation Extraction System". In: *Proceedings of the COLING/ACL on Main Conference Poster Sessions*. Sydney, Australia: Association for Computational Linguistics, pp. 667–674 (cit. on p. 5).

Schiller, Anne, Simone Teufel, and Christine Thielen (1999). "Guidelines für das Tagging deutscher Textcorpora mit STTS". In: *Universität Stuttgart and Universität Tübingen* (cit. on p. 26).

Schmitz, Michael et al. (2012). "Open Language Learning for Information Extraction". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea, pp. 523–534 (cit. on pp. 6, 8, 13, 23, 29, 30, 40, 44, 45).

Schneider, Gerold, Michael Hess, and Paola Merlo (2008). "Hybrid Long-Distance Functional Dependency Parsing". PhD thesis. Zürich, Switzerland: University of Zürich (cit. on p. 38).

Schuler, Karin Kipper (2005). "VerbNet: A broad-coverage, comprehensive verb lexicon". PhD thesis. Philadelphia, PA, USA: University of Pennsylvania (cit. on p. 23).

Sennrich, Rico and Beat Kunz (2014). "Zmorge: A German Morphological Lexicon Extracted from Wiktionary". In: *Proceedings of the 9th International Conference on Language Resources and Evaluation*. Reykjavik, Iceland, pp. 1063–1067 (cit. on p. 38).

Sennrich, Rico, Gerold Schneider, et al. (2009). "A New Hybrid Dependency Parser for German". In: Potsdam, Germany, pp. 115–124 (cit. on p. 38).

Sennrich, Rico, Martin Volk, and Gerold Schneider (2013). "Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis". In: *Proceedings of Recent Advances in Natural Language Processing*. Hissar, Bulgaria, pp. 601–609 (cit. on p. 38).

Sleator, Daniel Dominic and David Temperley (1995). "Parsing English with a Link Grammar". In: *CoRR* abs/cmp-lg/9508004 (cit. on pp. 7, 16).

Soderland, Stephen (1999). "Learning Information Extraction Rules for Semi-Structured and Free Text". In: *Machine Learning* 34.1-3, pp. 233–272 (cit. on p. 4).

Soderland, Stephen et al. (2010). "Adapting Open Information Extraction to Domain-Specific Relations". In: *AI Magazine* 31.3, pp. 93–102 (cit. on p. 11).

Srikant, Ramakrishnan and Rakesh Agrawal (1996). "Advances in Database Technology — EDBT '96: 5th International Conference on Extending Database Technology Avignon, France, March 25–29, 1996 Proceedings". In: Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. Mining Sequential Patterns: Generalizations and Performance Improvements, pp. 1–17 (cit. on p. 19).

Surdeanu, Mihai et al. (2008). "The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies". In: *Proceedings of the 12th Conference on Computational Natural Language Learning*. Manchester, UK, pp. 159–177 (cit. on p. 27).

Wang, Mingyin, Lei Li, and Fang Huang (2014). "Semi-supervised Chinese Open Entity Relation Extraction". In: *Proceedings of the 3rd IEEE International Conference on Cloud Computing and Intelligence Systems*. Shenzhen and Hongkong, China, pp. 415–420 (cit. on pp. 9, 13, 18, 33).

Weld, Daniel S. et al. (2008). "Intelligence in Wikipedia". In: *Proceedings of the 23rd National Conference on Artificial Intelligence*. Chicago, Illinois, USA, pp. 1609–1614 (cit. on p. 5).

Wu, Fei and Daniel S Weld (2010). "Open Information Extraction using Wikipedia". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, pp. 118–127 (cit. on pp. 7, 13–15, 18, 24, 29, 41).

Xavier, Clarissa Castellã and Vera Lúcia Strube de Lima (2014). "Boosting Open Information Extraction with Noun-Based Relations." In: *Proceedings of the 9th International Conference on Language Resources and Evaluation*. Reykjavik, Iceland, pp. 96–100 (cit. on pp. 9, 13, 20, 28, 44, 46).

Xu, Ying et al. (2013). "Open Information Extraction with Tree Kernels". In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia, USA, pp. 868–877 (cit. on pp. 13, 19, 22, 29, 31).

Yahya, Mohamed et al. (2014). "ReNoun: Fact Extraction for Nominal Attributes." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar, pp. 325–335 (cit. on pp. 9, 13, 24).

Yamada, Hiroyasu and Yuji Matsumoto (2003). "Statistical Dependency Analysis With Support Vector Machines". In: *Proceedings of 8th International Workshop on Parsing Technologies*. Nancy, France, pp. 195–206 (cit. on p. 34).

Zhila, A and Alexander Gelbukh (2013). "Comparison of Open Information Extraction for English and Spanish". In: *Proceedings of the 19th Annual International Conference Dialog 2013*. Bekasovo, Russia, pp. 714–722 (cit. on pp. 9, 13, 20, 32).

Zhu, Jun et al. (2009). "StatSnowball: a Statistical Approach to Extracting Entity Relationships". In: *Proceedings of the 18th International World Wide Web Conference*. Madrid, Spain, pp. 101–110 (cit. on pp. 8, 13, 30).

Zielinski, Andrea, Christian Simon, and Tilman Wittl (2009). "State of the Art in Computational Morphology: Workshop on Systems and Frameworks for Computational Morphology, SFCM 2009, Zurich, Switzerland, September 4, 2009. Proceedings". In: Berlin, Germany: Springer Berlin Heidelberg. Chap. Morphisto: Service-Oriented Open Source Morphology for German, pp. 64–75 (cit. on p. 38).