Boosting Relation Extraction
with Limited Closed-World Knowledge

Studienarbeit
von

Sebastian Krause

am Institut für Informatik
der Humboldt-Universität zu Berlin

Betreuer: Prof. Dr. Ulf Leser
Betreuer: Dr. Feiyu Xu, DFKI GmbH

Bearbeitungszeit: Januar 2010 – April 2010
Abstract

A number of different approaches to the acquisition of relation extraction rules exist. In contrast to supervised learning algorithms and manual approaches, using a minimally supervised method requires only a small number of hand-crafted examples in order for a learning algorithm to start. Unfortunately, minimally supervised machine-learning systems, such as DARE, are faced with the problem of learning wrong rules, which lead to faulty precision in the extraction task. However, there is a lot of domain-specific knowledge available for many target relations that can be exploited for validation of learned results. This student research project proposes and implements strategies for rule confidence estimation on top of the existing learning system DARE, by taking additional domain knowledge into account. The knowledge is re-structured to form closed worlds, which gives DARE the ability to reason about the correctness of extracted relation instances and learned rules during the learning process. This information is included into the rule ranking to boost the rating of good performing rules and therefore increase the overall extraction precision. Two ranking methods are developed and evaluated. Comparative experiments are conducted that prove the benefit of the proposed strategies. The performance in terms of reliability of rule confidence estimation and relation instance precision is improved, although there is also a decrease in recall. Finally, another experiment with a different target domain is performed to investigate the domain adaptability of the approach. The experimental results in the second domain confirm that using closed-world knowledge is a promising approach to the improvement of the learning and extraction performance.
1 Introduction

In this thesis, rule confidence estimation strategies are investigated and implemented to improve the precision of an existing minimally supervised machine-learning system for Relation Extraction (RE), called **DARE** (Xu et al., 2007; Xu, 2007). The key idea of this work is to exploit previously existing information about RE target domains as closed-world knowledge to validate intermediate results of the learning process.

**Background** In general, a standard RE system extracts relation instances from natural language texts by applying extraction rules to those texts. Rules can be constructed either by human experts or by machine-learning systems. However, human experts are expensive and not always available for all domains or target relations. Therefore, machine-learning systems are important for real world applications and are very useful for supporting the domain adaptability of RE systems. Minimally supervised machine-learning techniques need only limited initial knowledge and minimal human intervention, while showing promising results, cf. Xu et al. (2007). For example, many of these systems include iterative learning algorithms that only need a small number of *seed* examples as input, which are either examples for correct patterns of the target relation or example instances of the relation. Systems that implement such an iterative learning algorithm are presented by Agichtein and Gravano (2000, *Snowball*), Brin (1998, *DIPRE*), Etzioni et al. (2005, *KnowItAll*), Pantel and Pennacchiotti (2006, *Espresso*), Siniakov (2008, *GROPUS*), Stevenson and Greenwood (2005), Yangarber et al. (2000, *ExDisco*) and Xu et al. (2007, **DARE**). Those systems show very promising results, but are all faced with the same problem, namely the learning of wrong rules. Thus a way of estimating the confidence of an induced rule is needed.

**Contribution** This thesis describes an approach to rule confidence estimation that exploits additional knowledge about the domain of the target relation. The additional knowledge is constructed as a set of closed worlds that allow reasoning about the correctness of the created extraction rules. This is achieved by implicitly generating wrong instances of the target relation. Two rule ranking methods are developed in this thesis that make use of the closed-world validation mechanism.

The rule ranking methods are evaluated using the Relation Extraction system **DARE** (*Domain Adaptive Relation Extraction*), which is described by Xu et al. (2007) and Xu (2007). **DARE** takes a small set of example instances of a target relation as its input and generates extraction rules by searching for sentences containing them in the supplied text corpus. The newly created extraction rules are then applied to the corpus again to extract new relation instances, which in turn are utilized to generate new rules. **DARE** provides an initial approach to rule ranking which is mainly dependent on the relevance values of the terms occurring in the rules. It presumes a domain term database assigned with relevance values. The integration of closed-world knowledge is on the one hand independent of domain term resource and on the other hand provides very reliable predication of confidence.
Evaluation experiments are conducted for two different target relations. The first one describes the award winning event of Nobel prize laureates, i.e. it relates persons with prizes, prize areas and years. The second relation consists of persons that are married to one another, as well as of a year in which the marriage took place. For the two target relations two different text corpora are used, both made up of newspaper articles from recent years.

The rest of this work is structured as follows. Section 2 (Domain Adaptive Relation Extraction: \textit{DARE}) describes the task of RE and the employed system \textit{DARE}. Section 3 (Related Work) explains other published approaches to rule confidence estimation for relation extraction, while Section 4 (Approach) presents this thesis’ approach. Section 5 (Evaluation) reports the performed experiments and their results. Section 6 (Conclusion) gives a summary of this thesis.
2 Domain Adaptive Relation Extraction: DARE

This section starts with a short introduction of the Information Extraction research area of which Relation Extraction is a part of. Afterwards, a detailed description of the DARE system is presented.

2.1 Relation Extraction in Context of Information Extraction

Relation Extraction (RE) is a subfield of the research area of Information Extraction (IE). According to Carstensen et al. (2009), the goal of IE is to find relevant information about a specific target domain in natural language texts and to structure this information, while ignoring irrelevant parts of the text. In contrast to Information Retrieval, a deeper analysis of text is performed, i.e. only the parts of the text documents that contain relevant information are extracted. What kind of information is relevant is application dependent and must be defined at the beginning of an IE task. An IE system can be very complex, Carstensen et al. (2009) have identified three essential subtasks:

- **Named Entity Recognition**
- **Coreference Resolution**
- **Relation Extraction**

**Named Entity Recognition** The task of NER is to recognize text strings in the plaintext documents which denote proper names of entities (→ named entities) and to classify them to their semantic types, e.g. persons (Barack Obama), locations (Berlin, England, etc.), organizations (Microsoft) or prizes (Pulitzer prize).

Figure 1 illustrates the task of NER by showing a short text sample in which some named entities were manually marked together with the corresponding entity type.

**Coreference Resolution** This task is concerned with identifying and resolving those phrases and named entities that refer to the same entities. As an example, consider a text introducing a person at first, usually by stating the full name of the person, and then
President Barack Obama spends his 49th birthday on Wednesday without his family while First lady Michelle Obama is vacationing in Spain. Last year Obama was honored with the 2009 Nobel prize for Peace.

**Figure 2:** Example text in which all references to the entity Barack Obama are marked.

\[
\langle \text{Barack Obama} , \ 2009 , \ \text{Nobel} , \ \text{Peace} \rangle
\]

**Figure 3:** Relation with instance: Award winning event of prize laureates.

referring to this person later on in various ways, e.g. by using personal pronouns or (e.g. shorter) variants of the name. This is illustrated in Figure 2, in which all highlighted phrases refer to same entity, namely Barack Obama. Note that in this example Michelle Obama is not marked as this named entity does not refer to Barack Obama.

**Relation Extraction** The goal of RE is to identify those *semantic* relations between the recognized entities that are mentioned in the text. Stated differently, the goal is to extract *tuples* of named entities from the text, when there is textual evidence that those entities are related in a specific way. In traditional Information Extraction, the relations that are to be recognized are called *target relations*. The slots of a target relation (*arguments*) are filled with named entities of a certain *type*. As an example, consider the 4-ary relation from Figure 3 whose tuples express the award winning event of prize laureates. The instance shown in this figure was extracted from the text sample in Figure 1. The type of the argument *Winner* is *person*, the types of the other three arguments *PrizeYear*, *PrizeName* and *PrizeArea* are *year*, *prize* and *prize area*, respectively. This means that only named entities of the types *person*, *year*, *prize* and *prize area* can be related in the sense of the target relation.

**2.2 The DARE System**

The *DARE* system is made up of two main parts, as depicted in Figure 4. The *classifier*, *rule learning* and *relation extraction* components (lower part of the diagram) work on a linguistically preprocessed version of the text corpus. The corresponding text preprocessing (upper part in the figure: *linguistic annotation*) is performed at the very beginning of the system’s operation. The output of this step is an annotated corpus that incorporates information about contained named entities, the grammatical structure of sentences as well as linguistic properties of words. This data is exploited in all other components of the system, namely the ones that use supplied *seeds* for rule learning on the annotated corpus and learned rules for relation extraction from the corpus.
Figure 4: Architecture of the DARE System. (From http://dare.dfki.de, retrieved 10th August 2010)
Mohamed ElBaradei won the 2005 Nobel Prize for Peace on Friday because of ... 

Figure 5: Example sentence containing a mention of the prize award event relation.

The next subsection describes the linguistic annotation step and the corresponding tools used in the experiments of this thesis. The learning and extraction process is explained in detail in Subsection 2.2.2. Afterwards, the present rule ranking strategy is discussed.

2.2.1 Linguistic Annotation

NER & Coreference Resolution During the preprocessing of the plain-text corpus, a number of Natural Language Processing (NLP) tools are employed. OpenCalais\(^1\), a web service that offers to enrich submitted plain text with semantic metadata, was employed for the basic NER as well as for Coreference Resolution. OpenCalais is able to recognize entities of a variety of types, e.g. person, organization and location, but also more exotic ones like medical condition. Across them, OpenCalais shows precision values of more than 90% and recall values from 60% to 70%, according to MacEachren et al. (2010) and Iacobelli et al. (2010). For this thesis’ setting, the only entity type used from OpenCalais is person. SProUT\(^2\), described by Drozdzynski et al. (2004), is able to recognize named entities and their types more flexible and in greater detail than OpenCalais does and is thus being used here to extend the named entity information from OpenCalais. For example, SProUT was modified to detect entities of type prize, prize area and year, which are missed out by OpenCalais.

Dependency Parsing The second important step of the linguistic annotation is the analysis of the grammatical structure of sentences. For this part, the Stanford Parser\(^3\), described by Klein and Manning (2003), is employed. The parser takes a single sentence as input and outputs the dependency relations between the words in the sentence, represented in a tree. This data is enriched with word stemming information from SProUT, e.g. is winning, wins and won are normalized to win. As an example for this dependency parsing consider the sentence from Figure 5 for which Figure 6 shows a simplified dependency tree. Information from NER and Coreference Resolution is already included. The tree states, for example, that won is the head of the sentence, as it is the root node of the tree. The subtree of the node Prize forms the object of the verb won, which assembles a subject-verb-object phrase together with the subject Mohamed ElBaradei. Furthermore, the object of this phrase is modified by the subtree of the node for. These so-called dependency relations are an integral part of DARE’s rule representation. For a description of all dependency types see the manual of the Stanford Parser written by de Marneffe and Manning (2008).

\(^1\)http://www.opencalais.com/
\(^2\)http://sprout.dfki.de/
\(^3\)http://nlp.stanford.edu/software/lex-parser.shtml
2.2.2 \textit{DARE} Learning Algorithm

This subsection explains the learning algorithm of \textit{DARE}. The involved system components are shown in the lower part of Figure 4. The examples are taken from (Xu, 2007).

\textbf{From seeds to rules} The learning process starts with a number of example instances of the target relation, called \textit{seeds}. The classifier’s task is to identify those sentences in the corpus that contain arguments of the seeds. This is implemented using the text search engine Lucene\textsuperscript{4}, taking the seed arguments as a free text query. See for example the relation instance in Figure 7. Its arguments are Mohamed ElBaradei, 2005, Nobel and Peace. When using this instance as a seed, i.e. transforming the arguments into a Lucene query, the classifier component will return all sentences from the corpus that textually contain the arguments. Because words can belong to different entity types (Nobel can be a prize or a person), sentences whose argument occurrences do not match the entity types of the seeds are removed. For example, sentences containing the person Alfred Nobel are removed because the PrizeName argument Nobel of the seed of Figure 7 is a prize, not a person. An example from the remaining set of sentences is the sentence shown in Figure 5.

The assumption of \textit{DARE}’s learning algorithm is that all the sentences from the classifier express the target relation between the arguments of the seed. For the example seed this means that all the sentences from the classifier mention the \textit{Award winning event of prize laureates} relation between the entities Mohamed ElBaradei, 2005, Nobel and Peace. This semantic information is reflected in the grammatical structure of the sentences. \textit{DARE}’s “view” on the grammar of sentences are dependency

\footnote{\url{http://lucene.apache.org/}}
relations between words, represented in a dependency tree, as already mentioned in Section 2.2.1 (Linguistic Annotation). The task of pattern extraction (see Figure 4) is to create dependency-tree templates, called patterns or rules, from the sentences the classifier returned. These templates will then be useful to identify sentences in the corpus that contain other instances of the target relation.

Patterns are extracted from the parts of the sentence’s dependency tree that contain the seed’s arguments. The first step is to replace the nodes in the tree containing the seed arguments by the corresponding argument names from the seed. The dependency for the sentence from Figure 5 is given in Figure 6. The first step of pattern extraction will transform the dependency tree to the intermediate tree (or intermediate pattern) shown in Figure 8.

The next step is the actual pattern creation, which is performed bottom up, i.e. the intermediate tree’s lowest subtrees are transformed to rules first. The lowest subtree of Figure 8 is the subtree under the node for. This subtree states only that the word for is in the dependency relation “pcomp-n” with the PrizeArea argument of the target relation. DARE’s rules represent this information in the form of (nested) attribute-
The DARE rule “area” for the current subtree is shown in Figure 9. “head” and “daughter” refer to nodes of a tree, i.e. to the root node for and its child node PrizeArea. The value of the attribute “head” is itself a list of attribute-value pairs. They state that the node has the fixed lexical form for, meaning it is non-variable and cannot be substituted by other words. Furthermore, the node has the part-of-speech tag preposition. This information is taken from the NLP analysis of the sentence, just like the dependency relations. The rule further says that the word which is in the dependency relation “pcomp-n” with the root node is the PrizeArea argument. This rule is quite simple as it is only capable of extracting or detecting a single argument of the target relation in a sentence.

After the creation of the “area” rule, the subtree of the node Prize is transformed to a rule. This rule, named “year_prize_area” and shown in Figure 10, is able to extract three arguments of the target relation and is constructed using the rule “area”. Its meaning is analogous to the previously described rule, apart from the third “daughter” attribute. Here, the value is a reference to the rule “area”, which is responsible for the treatment of the subtree that is in dependency relation “mod” with the root node of the current subtree.

Finally, the rule for the top node of the intermediate tree from Figure 8 is created. The corresponding rule “winner_year_prize_area” is depicted in Figure 11. This rule again contains a reference to another rule, namely the rule from Figure 10. As the pattern extraction is done for all sentences that the classifier component returned, the result of the pattern extraction is a vast amount of rules. Because of this, grouping of rules, deletion of redundant rules and compression of similar rules is performed, denoted in Figure 4 by rule induction.

From rules to new instances After the rule learning took place, the newly created rules are applied to the linguistically annotated corpus, i.e. to the dependency trees of the sentences. Applying a rule to a dependency tree means to check if the tree matches the fixed parts of the rules, i.e. whether the nodes of the tree match the part-of-speech tags, the entity types and the non-variable words and whether the edges of the tree (the dependency relations) match the ones denoted in the rule. If a tree satisfies a
Rule name :: year_prize_area
Rule body ::

\[
\begin{align*}
\text{head} & : \begin{cases} \text{pos} & : \text{noun} \\
\text{lex-form} & : \text{Prize} \end{cases} \\
\text{daughter} & : \begin{cases} \text{lex-mod} & : \text{PrizeYear} \\
\text{rule area} & : <\text{PrizeArea}> \end{cases} \\
\text{daughter} & : \begin{cases} \text{lex-mod} & : \text{PrizeName} \end{cases} \\
\text{daughter} & : \begin{cases} \text{mod} & : \begin{cases} \text{head} & : \begin{cases} \text{pos} & : \text{preposition} \\
\text{lex-form} & : \text{for} \end{cases} \\
\text{rule area} & : <\text{PrizeArea}> \end{cases} \end{cases}
\end{align*}
\]

Output :: \((\text{PrizeYear}, \text{PrizeName}, \text{PrizeArea})\)

Figure 10: Pattern extraction step 2: Rule 2.

Rule name :: winner_year_prize_area
Rule body ::

\[
\begin{align*}
\text{head} & : \begin{cases} \text{pos} & : \text{verb} \\
\text{mode} & : \text{active} \\
\text{lex-form} & : \text{win} \end{cases} \\
\text{daughter} & : \begin{cases} \text{subject} & : \text{Winner} \end{cases} \\
\text{daughter} & : \begin{cases} \text{object} & : \begin{cases} \text{head} & : \begin{cases} \text{pos} & : \text{noun} \\
\text{lex-form} & : \text{Prize} \end{cases} \\
\text{rule year_prize_area} & : <\text{PrizeYear, PrizeName, PrizeArea}> \end{cases} \end{cases} \\
\end{align*}
\]

Output :: \((\text{Winner}, \text{PrizeYear}, \text{PrizeName}, \text{PrizeArea})\)

Figure 11: Pattern extraction step 2: Rule 3.
\[ \text{confidence}(\text{rule}) = \beta^i \cdot |\text{fromSeeds}(\text{rule})| \cdot |\text{toSentences}(\text{rule})| \]

where

\[ \beta : 0.8 \]
\[ i : i\text{-th iteration of the learning algorithm extracted seeds} \]
\[ \text{fromSeeds}(\text{rule}) : \text{ancestors seeds of rule} \]
\[ \text{toSentences}(\text{rule}) : \text{sentences from corpus that matched rule} \]

**Figure 12:** Rule confidence estimation of used *DARE* version.

rule, the nodes of the tree at the positions of the arguments in the rule are extracted, thus creating a new relation instance. More complex rules, i.e. those containing more arguments, are preferred to simpler ones. The results of this relation extraction step are relation instances that can be used as new seeds. These seeds start a new cycle of the learning algorithm that again creates more rules and again new relation instances. This interplay continues until no new relation instances are extracted, in which case the learning algorithm stops.

### 2.2.3 Ranking

The *DARE* version that is used as a baseline in this thesis incorporates a simple rule ranking. The score of a rule is on the one hand dependent on the number of times it was used to extract new instances. On the other hand, the trustworthiness of its origin (i.e. seeds/relation instances) is also taken into account, here approximated by combining the number of the seed instances with the iteration cycle of the learning algorithm in which they were extracted. Figure 12 depicts a formula formalizing the interplay of these rule properties. The hope is that the higher the calculated confidence value of a rule is, the higher its quality will be. This is based on the assumption that patterns which often occur in the text corpus are likely to be correct. At least for a corpus whose sentences are more likely to contain a mention of the target relation than arbitrary sentences, this is a reasonable assumption. Such a bias for a corpus can be achieved, e.g., by utilizing an Information Retrieval component for the corpus creation.

### 2.2.4 Overview of the Learning Algorithm

Figure 13 summarizes the learning algorithm of *DARE*. The variable \( i \) denotes the current iteration of the loop started in line 4. This loop is processed as long as there are instances that were not already exploited as *seeds*. The function \( \text{getRules}(\text{seeds}) \) creates extraction patterns from the sentences in the corpus the passed instances appear in. Using the provided extraction rules, \( \text{getInstances}(\text{rules}_i) \) extracts instances from the corpus. Finally, \( \text{rank}(\text{rules}_i) \) scores the rules \( \text{rules}_i \) by applying the confidence
estimation function from Figure 12. The rule score is not exploited in the learning algorithm itself, but it can be used for increasing precision by introducing a threshold for the score of a rule. This way, rules and their extractions are excluded if their estimated correctness is too low, i.e. if the score is below the threshold.

2.3 The Problem of Relation Extraction

When applying an iterative approach to rule generation, each cycle of the learning algorithm produces rules whose distance to the initial trustworthy seeds grows bigger. This might lead to the creation of wrong rules, i.e. rules that no longer express the target relation and thus to the extraction of incorrect relation instances from the text documents. Xu (2007) performs an analysis of the rules created by a run of DARE for the award winning event of prize laureates relation. The rules are divided into four groups: good, useless, dangerous and bad. Good rules extract only correct instances, their counterpart are bad rules, which extract only incorrect instances. Dangerous rules extract both correct and incorrect relation instances and the useless ones do not extract any (new) instances. In the performed evaluation, the major part (83%) of the rules were useless, mainly because of being too specific to match other sentences than those they were originating from. 11.7% of the rules were classified as good, 1.6% as bad and 3.7% as dangerous.

Xu (2007) also examined the impact of bad rules on the performance of DARE. Four different error sources for extracted incorrect relation instances were identified. The major part of the errors (55.9%) is caused by wrong Natural Language Processing, i.e. by erroneous NER, Coreference Resolution and Dependency Parsing of the sentences.
29.4% of the errors account to either wrong facts expressed in the corpus or to facts mentioned in context of modality or negation. The remaining 14.7% of the wrong relation instances are caused by problematic rules. Thus, trying to improve the quality of rule learning seems to be a promising way to improve the precision of RE systems.

A particular problem of DARE’s approach to RE is the existence of neighboring (semantically related) relations to many target relations. For example, when choosing the award winning event of prize laureates as the target relation, the relation consisting of persons being nominated for a prize constitutes a superset. The rule learning algorithm drifts to this relation by first extracting a correct relation instance for the target relation, i.e. an instance containing a person who was indeed awarded a certain prize. This instance is then exploited as a seed for the creation of rules. If the nomination of the person is also mentioned in the corpus, patterns expressing the nomination event, instead of the actual awarding event, are learned. This is called semantic drift. This phenomenon is caused by the learning algorithm’s assumption that all sentences, which contain the arguments of an instance of a relation, indeed express this relation. Another example for a potentially problematic relation pairing is the marriage relation between persons and the more general relation stating that people met each other at some occasion. Assume a relation instance for the marriage relation between Nicole Kidman and Tom Cruise was extracted. If there are sentences in the corpus mentioning that those two people met e.g. at a charity performance, the rule generation step of the learning process will lead to the creation of patterns expressing only that two persons meet, and not that they are married.

Because of the potential incorrectness of learned rules, RE systems usually perform a rule validation, rule ranking or rule confidence estimation to identify dangerous and bad rules. Systems that implement such a behavior are presented by, e.g., Agichtein and Gravano (2000, Snowball), Brin (1998, DIPRE), Etzioni et al. (2005, KnowItAll), Pantel and Pennacchiotti (2006, Espresso), Plake et al. (2006, AliBaba), Siniakov (2008, GROPUS), Stevenson and Greenwood (2005) and Yangarber et al. (2000, ExDisco).

This thesis presents an approach were additional domain knowledge structured as closed worlds is exploited, which is described in detail in Section 4 (Approach). The rule confidence estimation of the used DARE version was already described in Section 2.2.3 (Ranking). Other published approaches are explained in Section 3 (Related Work).
3 Related Work

In existing Relation Extraction systems different approaches to the identification of wrong rules and to rule confidence estimation are employed.

**Duality Principle** When using an iterative approach for the learning process, the *Duality Principle* is a common guideline for the estimation of the correctness of a rule. The Duality Principle, described by Brin (1998) and Yangarber (2001), states that the confidence value of learned rules is dependent on the confidence of their origin in the learning process. For example, if \( r \) is extracted from a set \( D \) of documents, \( r \)'s confidence score is dependent on the target-domain relevance of the documents in \( D \). Sudo et al. (2003) and Yangarber (2001) utilize this principle for their work on relation extraction. In their systems, rules stemming from highly domain relevant documents have a higher confidence value than those created from less relevant documents. Agichtein and Gravano (2000), Brin (1998), Pantel and Pennacchiotti (2006) and Xu et al. (2007) extract rules from sentences that contain a mention of already known instances of the target relation. Following the duality principle, they score a rule dependent on the confidence score of those known instances. The duality-principle strategy is often enhanced by including frequency information into the confidence calculation. This is usually implemented by increasing the score of frequently used rules, i.e. rules that stem from many sentences or rules that extract a lot of relation instances.

**Constraints & Domain-Relevant Words** Agichtein (2006) presents the idea to define certain constraints for possible values within a relation instance or combinations of them. These constraints have to be fulfilled, in order for an instance to be correct. They can be exploited to prevent the Relation Extraction system from creating wrong rules and extracting wrong instances. Another approach to improve the quality of rule confidence estimation is to calculate the target domain relevance of the terms that occur in extraction patterns, which is done by Xu (2007). The domain relevance of a term is calculated by relating the frequency of it in domain relevant documents to its frequency in texts of other domains, i.e. documents not dealing with the domain of the target relation. The more relevant terms a pattern contains and the higher their relevance score is, the higher is the calculated confidence of the pattern.

**Simple Rule Filters** Nguyen et al. (2010) propose several simple techniques for filtering sets of generated extraction patterns. Their strategies make use of rule and corpus properties, like the complexity and length of rules and text documents. Similar to Xu (2007), they learn domain-relevant *trigger words* and apply them for rule confidence estimation. The conducted evaluation proves that even simple heuristics can help to determine the rules with the worst extraction performance.

**Cosine Similarity** Stevenson and Greenwood (2005) proposed an approach to Relation Extraction that starts with a number of positive example patterns, i.e. patterns that
definitely express the target relation. When processing the input text documents and their grammatical sentence structure, their algorithm tries to identify patterns with a similar meaning to those rules already known. This is achieved by exploiting information from WordNet (Fellbaum, 1998), an ontology of the words of the English language. Their approach is based on the assumption that useful patterns will have similar meanings to the already accepted patterns. Using the WordNet data, they apply the cosine metric for similarity calculation, hence the name of this idea.

**Negative Rules and Examples** In addition to rule confidence estimation, the learning of negative rules, i.e. rules that do not belong to the target relation, is useful. Etzioni et al. (2005) and Yangarber (2003) employ the so-called Counter-Training for detecting negative rules. This is done by learning patterns for multiple domains or target relations at the same time, where then positive patterns of a certain domain serve as negative examples for the other domains. Bunescu and Mooney (2007) follow a classification-based approach to Relation Extraction, in which they use both positive and negative sentences for a target relation as input to a Support Vector Machine classifier. Uszkoreit et al. (2009) present different approaches on how to learn negative rules from negative relation instances by employing Relation Extraction strategies. Identified negative rules are then integrated into the “normal” rule learning and relation extraction process to increase the precision of the resulting instances.

**Proposed Approach** The main idea of this thesis’ approach is to exploit additional knowledge about the target domain directly for rule confidence estimation. The additional knowledge is given as a set of correct instances of the target relation. By creating closed worlds for certain argument values of the target relation, the correctness of some of the learned instances becomes determinable. In contrast to other seed-based RE systems, e.g. DARE by Xu et al. (2007) or Snowball by Agichtein and Gravano (2000), the given instances are not only used to initialize the learning process, but also for ranking the rules. This way, estimation of rule confidence is more accurate and less reliant on heuristics.

In contrast to the Counter-Training approach and to the acquisition of negative rules, no learning of non-target relations is necessary. The proposed strategy is also favorable to the constraint strategy of Agichtein (2006) because the lack of domain adaptability is avoided. Their constraints are hand-crafted and must be created for each domain on its own by intellectual work. In contrast, the additional knowledge in this thesis’ strategy can be created automatically or is even already existing.
4 Approach

This section describes the new approach to rule confidence estimation that is proposed in this thesis.

4.1 Idea

The main problem of rule learning is the learning of wrong rules, namely, rules that do not express the target relation explicitly. The idea of this thesis is to exploit already existing knowledge about the domain of the target relation for the identification of those wrong rules. The additional knowledge is given as a set of instances of the target relation. For all of these instances certain conditions are fulfilled. These conditions allow the Relation Extraction (RE) system to decide for some of the learned instances whether they are correct or not, i.e. whether they are part of the target relation. This information is then exploited for rule confidence estimation.

4.1.1 Definitions

Section 2.1 (Relation Extraction in Context of Information Extraction) introduced the term named entity. The type of a named entity is its semantic class, e.g. the type of the named entity Barack Obama is person, the type of Microsoft is organization and so on.

Definition 1 (Relation). Let \( t \) be a named-entity type and let \( \mathcal{NE}_t \) be the set of all named entities of type \( t \). Let \( T \) be a bag of named-entity types and let \( n = |T| \). Then any set \( R \) with

\[
R \subseteq \prod_{t \in T} \mathcal{NE}_t
\]

is called an \( n \)-ary relation. An element \( \text{inst} \in R \) is called an instance of this relation. An element of \( \text{inst} \) is called (relational) argument.

An example for a relation is the following set:

\[
R_{\text{example}} = \{(\text{Barack Obama}, \text{Microsoft}), (\text{Marco Rima}, \text{Nestle})\}
\]

Here, \( R_{\text{example}} \subseteq \mathcal{NE}_{\text{person}} \times \mathcal{NE}_{\text{organization}} \). Note that Definition 1 does not require a relation to have any reasonable semantics. But because RE extracts information from natural language texts, it is reasonable to choose only those relations as target ones that have a clearly defined meaning. An example for such a meaningful relation is the 4-ary prize award event relation:

\[
R_{\text{prize}} \subseteq \mathcal{NE}_{\text{person}} \times \mathcal{NE}_{\text{year}} \times \mathcal{NE}_{\text{prize}} \times \mathcal{NE}_{\text{area}}
\]

Naturally, \( R_{\text{prize}} \) should only contain those instances that refer to a real-world prize awarding. Since every target relation is assumed to have a meaning, the relational arguments are usually named for better readability. In case of the prize award event relation these argument names are: Winner, PrizeYear, PrizeName and PrizeArea. A concrete value of an argument, i.e. a named entity like Barack Obama, is referred to as the argument value.
Definition 2 (Constraint). Let $\mathcal{R}$ be an $n$-ary relation with $\mathcal{R} \subseteq N \times \ldots \times N$. Then any pair $(i, v)$ with $i \in \mathbb{N}, 1 \leq i \leq n$ and $v \in N$ is called a constraint for $\mathcal{R}$.

Definition 3 (Fulfillment of Constraints). Let $\mathcal{R}$ be an $n$-ary relation and let $\text{inst} = (v_1, \ldots, v_n) \in \mathcal{R}$. Let $(i, v)$ be a constraint for $\mathcal{R}$ and let $C$ be a set of constraints for $\mathcal{R}$. $\text{inst}$ fulfills $(i, v)$, iff $v_i = v$. $\text{inst}$ fulfills $C$, iff $\forall (i, v) \in C : v_i = v$.

$(3, \text{Nobel})$ is a constraint for $\mathcal{R}_{\text{prize}}$. It is fulfilled for all instances of $\mathcal{R}_{\text{prize}}$ that describe the awarding of a Nobel laureate. For better readability, the first element of a constraint can be substituted by the corresponding argument name. For example, instead of $(3, \text{Nobel})$, $(\text{PrizeName}, \text{Nobel})$ can be used.

Definition 4 (Closed World). Let $C$ be a set of constraints for the $n$-ary relation $\mathcal{R}$. Let $\mathcal{W} \subseteq \mathcal{R}$. Then the pair $(\mathcal{W}, C)$ is called a closed world, iff \( \forall \text{inst} \in \mathcal{R} : (\text{inst} \text{ fulfills } C \implies \text{inst} \in \mathcal{W}) \)

A set of closed worlds is called a closed-world knowledge database.

From now on, closed world is abbreviated by $\text{cw}$ and closed-world knowledge database by $\text{cwDB}$.

Consider the binary corporate acquisition relation $\mathcal{R}_{\text{corp}}$ between entities of type company: $\mathcal{R}_{\text{corp}} \subseteq N \times N$. We call the first argument Buyer, and the second one Acquisition. A $\text{cwDB}$ could then be made of the following closed worlds:

- $(\mathcal{W}_1, C_1)$ with $C_1 = \{ (\text{Buyer}, \text{Oracle}) \}$: $\mathcal{W}_1$ is a subset of $\mathcal{R}_{\text{corp}}$ that contains all the corporate-acquisition relation instances in which Oracle bought another firm.
- $(\mathcal{W}_2, C_2)$ with $C_2 = \{ (\text{Acquisition}, \text{Grundig}) \}$: $\mathcal{W}_2$ must contain all instances of $\mathcal{R}_{\text{corp}}$ in which Grundig was bought by another corporation.

This $\text{cwDB}$ allows us to infer about the correctness of certain extracted instances for the corporate acquisition relation. If a RE system extracted the instance $\langle \text{Oracle, Microsoft} \rangle$, we would be able to deduce that this instance is wrong, because it is not contained in $\mathcal{W}_1$. Thus, the $\text{cwDB}$ implicitly generates the wrong instance $\langle \text{Oracle, Microsoft} \rangle$. Note that if the RE system extracted the instance $\langle \text{Apple, IBM} \rangle$, we would not be able to infer about the correctness of this instance, as there is no corresponding $\text{cw}$ contained in the $\text{cwDB}$. The exact formalism for deducing the correctness of extracted instances is described later in Section 4.2 (Rule Ranking with Closed-World Knowledge).

For more complex relations, like the 4-ary prize award event relation, it is reasonable to reduce the size of the $\text{cwDB}$ by using more than one constraint for the $\text{cw}$. A possible $\text{cwDB}$ could e.g. contain a closed world $\text{cw}_{\text{Nobel, Peace}}$ with all Nobel prize laureates in the area Peace:

- $\text{cw}_{\text{Nobel, Peace}} = (\mathcal{W}, C)$ with $C = \{ (\text{PrizeName, Nobel}), (\text{PrizeArea, Peace}) \}$.

This $\text{cwDB}$ (more precisely: $\mathcal{W}$) would contain all instances with the instantiation values Nobel in the argument PrizeName and Peace in the argument PrizeArea, respectively.
4.1.2 Generation and Usage of cwDBs

There are two options for the creation of cwDBs: exploitation of existing knowledge databases and creation by hand.

When choosing the manual approach, the amount of human labor that has to be invested should be reasonably small. Thus, the size of a cw should be quite small. A possibility to achieve this is to make a clever choice of the relational argument that carries the closed-world property. Using arguments that are filled with entities of type person, like Winner in the prize awarding event relation, is an example, since only all the information about a single person has to be supplied this way. Other examples include arguments with (past) years or with (no longer existing) organizations.

Existing sources for structured knowledge on the Internet contain projects such as YAGO presented by Suchanek et al. (2007), DBpedia described by Bizer et al. (2009) and FreeBase introduced by Bollacker et al. (2008) and Kochhar et al. (2010). Furthermore, in some areas, such as Business Intelligence, there is nearly complete knowledge already present for recent years, while the task is to extract information only from recent news articles. So there is plenty of information to be used for our closed-world approach for many types of relations.

Although the relation instances contained in a closed-world knowledge database created in one of these ways could also be used as seeds for the learning process, the main idea of this thesis is to design a rule confidence estimation strategy that utilizes this information. Just increasing the number of seeds not necessarily improves the performance of an iterative RE system. In particular, Xu (2007) and Uszkoreit et al. (2009) state that domains can exhibit the small-world property, meaning that there exists a high connectivity between learned rules and extracted instances. In such cases, nearly all mentions of the target relation in the corpus can be reached in a few learning steps, almost regardless of the used seed instances.

4.2 Rule Ranking with Closed-World Knowledge

This section explains the usage of the closed-world knowledge database in DARE. The ranking steps described here are performed at the end of each learning cycle, i.e. at line 8 of Figure 13. At first the validation of extracted relation instances is described. Afterwards two different rule ranking methods are developed to investigate the best way of integrating the closed-world knowledge into rule confidence estimation. The exclusive ranking strategy excludes every pattern that extracted at least one wrong relation instance. The soft ranking is built on top of the duality principle as described in Section 3 (Related Work). It relaxes the exclusion of rules and takes other parameters such as the number of filled argument slots of instances, i.e. the specificity of instances, and the depth of the learning process into account.

Instance Validation Consider a RE setting with a given cwDB for a certain n-ary target relation R. The validation of extracted instances (insts) is performed as depicted in Figure 14. The intuition behind the first case is that each closed world cw in the
\[
\text{validate}(\text{inst}) = \text{correct} \iff \exists \text{cw} = (W, C) \in \text{cwDB} : \text{inst} \in W
\]
\[
\text{validate}(\text{inst}) = \text{wrong} \iff \exists \text{cw} = (W, C) \in \text{cwDB} : \text{inst} \notin W \\
\land \text{inst} \text{ fulfills } C
\]
\[
\text{validate}(\text{inst}) = \text{unknown} \iff \text{other cases}
\]

Figure 14: Instance validation with the \textit{cwDB}.

\[
\text{confidence}(\text{rule}) = \begin{cases} 
0 & \text{if } \exists \text{inst} \in \text{toInstances}(\text{rule}) : \text{validate}(\text{inst}) = \text{wrong} \\
1 & \text{otherwise.}
\end{cases}
\]

where

\text{toInstances}(\text{rule}) : \text{instances extracted from text with rule}
\text{validate}(\text{inst}) : \text{see Figure 14}

Figure 15: Exclusive ranking with the \textit{cwDB}.

\textit{cwDB} is basically just a set of relation instances with certain properties. Therefore, if a given instance is contained in a closed world, it will be \textit{correct}. The second case is the most interesting one. According to the definitions in the preceding subsection, if an instance fulfills all the constraints \(C\) of a closed world \(cw\), it shall be contained in the corresponding set \(W\). Now, if it is not contained, we will know it is definitely \textit{wrong} and not part of the target relation. The correctness of an examined instance \(\text{inst}\) is \textit{unknown} when there are no \(cws\) whose constraints \(C\) are fulfilled by \(\text{inst}\) or whose set \(W\) contains \(\text{inst}\). Additionally, in complex relations, there are often sentences in the text corpus that do not mention all of the arguments of the target relation, which results in underspecified relation instances, i.e. instances with in which some argument values are empty. This can also result in an \textit{unknown} correctness.

**Exclusive Ranking**  This ranking strategy makes naive use of the provided \textit{cwDB}. At the end of each learning cycle, all extracted instances are evaluated using the \textit{cwDB}. Then, depending on the result of this validation step, the rules that extracted these instances are ranked. There are just two possible scores for a rule: 0 or 1. Rules that extracted at least one wrong relation instance are scored 0, all other rules are assigned a score of 1. Figure 15 shows the ranking function.

**Soft Ranking**  This ranking strategy makes a softer usage of the information from the \textit{cwDB}. Although the confidence value of rules is still dependent on the truth value of their extracted instances, the extraction performance of a rule is calculated more
\[
\text{score}(\text{inst}) = \begin{cases} 
1 & \text{if } \text{validate}(\text{inst}) = \text{correct} \\
0 & \text{if } \text{validate}(\text{inst}) = \text{wrong} \\
U_{\text{inst}} & \text{if } \text{validate}(\text{inst}) = \text{unknown}
\end{cases}
\]

\[
U_{\text{inst}} = \sum_{\text{rule} \in \text{fromRules}(\text{inst})} \left( \frac{\sum_{j \in \text{fromSeeds}(\text{rule})} \text{score}(j)}{|\text{fromSeeds}(\text{rule})|} \right) \times \alpha \times \beta^i
\]

where
\text{validate}(\text{inst}) : \text{ see Figure 14}
\text{fromRules}(\text{inst}) : \text{ the set of rules that extracted } \text{inst}
\text{fromSeeds}(\text{rule}) : \text{ ancestors seeds of } \text{rule}
\alpha : \text{ specificity of } \text{inst}
\beta : 0.8
i : i-th iteration of the learning algorithm extracted \text{inst}

\textbf{Figure 16:} Instance scoring for the soft ranking with the cwDB.

precisely, compared to the binary exclusive ranking. All extracted instances of a bootstrapping cycle are again validated using the cwDB. As shown in Figure 16, each correct instance is given a score of 1, while each wrong instance is given a score of 0. If the correctness of an instance is unknown, its score will be dependent on the quality of the ancestors of the rules that extracted it. This approach follows the duality principle, i.e. implementing the assumption that good seeds create rules that in turn do a high quality extraction. \(U_{\text{inst}}\) calculates the confidence scores of those instances with unknown correctness. It calculates the average score of the seeds that created the rules which extracted the instance with unknown correctness. This value is then multiplied with the specificity of this instance as well as with a factor representing the higher potential of noise in later cycles of the learning algorithm. The specificity of an instance is important in complex relations, like the prize award event relation. For an n-ary target relation, it is calculated as the number of non-empty argument values of an instance divided by n:

\[
\text{specificity of instance } \langle v_1, \ldots, v_n \rangle : \frac{|\{ i \mid 1 \leq i \leq n \land v_i \text{ not empty} \}|}{n}
\]

Given the scores of the instances extracted by a certain rule, the rule’s confidence value is calculated as the average score of these values (Figure 17).
\[
\text{confidence}(\text{rule}) = \frac{\sum_{\text{inst} \in \text{toInstances}(\text{rule})} \text{score}(\text{inst})}{|\text{toInstances}(\text{rule})|}
\]

where

\text{toInstances}(\text{rule}) : \text{instances extracted from text with rule}

\text{score}(\text{inst}) : \text{see Figure 16}

\textbf{Figure 17}: Soft rule ranking with the \textit{cwDB}. 

21
5 Evaluation

This section contains information about the performed evaluation of the ranking strategies described in Section 4. As already explained, the DARE system is used as an implementation platform for the experiments.

5.1 Experimental Setting

To evaluate the impact on performance of Relation Extraction (RE) when using the additional closed-world knowledge, different experiments on two domains are performed. The first one is about the Nobel prize, while the second one deals with marriage relations between persons. For the Nobel prize domain, the Award winning event of prize laureates relation is used. The arguments for this relations are named the following way: Winner, PrizeYear, PrizeName and PrizeArea. The argument PrizeName is modified, so that it only takes the Nobel prize as a slot filler. Thus, the actual target relation of this domain is the Award winning event of Nobel prize laureates. The arguments can be filled with named entities of the following types: Winner with persons, PrizeYear with years, PrizeName with prizes (actually only Nobel) and PrizeArea with prize areas. This domain is the one that was used by Xu (2007), this way the results of this section can be compared to the ones described there. The corpus is made up of 3,328 newspaper documents from the Internet, all of them containing the word Nobel. From these documents 2,579 sentences containing at least named entities of types person and prize are selected to serve as the base corpus for the learning process. The closed-world knowledge is created from data available at http://nobelprize.org, i.e. the cwDB is made up of Nobel laureates of certain areas such as peace or literature. Two particular sizes of closed-world knowledge databases are tested. The complete cwDB is made up of Nobel laureates of all areas, serving as kind of a baseline showing the maximum possible performance of closed-world ranking. Formally, the complete cwDB is the set

\[
cwDB_{\text{complete}} = \{cw_{\text{Peace}}, cw_{\text{Literature}}, cw_{\text{Physics}}, cw_{\text{Medicine}}, cw_{\text{Economics}}, cw_{\text{Chemistry}}\}
\]

where \(cw_i = (W_i, C_i)\) and \(C_i = (\text{PrizeArea}, i)\) and \(W_i\) contains the relation instances of all Nobel awards in area \(i\). The small-sized cwDB is made up of all the laureates in the area Peace, i.e. there is only the closed world \(cw_{\text{Peace}}\) in this cwDB:

\[
cwDB_{\text{small}} = \{cw_{\text{Peace}}\}
\]

For the second domain, the marriage relation between persons is the chosen target relation. The arguments are NameOfSpouse, NameOfSpouse and Year. Year can be filled with named entities of type year, which means that the marriage relation between the NameOfSpouses, named entities of type person, was “active” in this year. Hence, sentences mentioning the divorce of two persons are also regarded as a match for this relation. Since the marriage relation is symmetric, the order of persons in the first two arguments is not relevant. The corpus is created from 6,850 popular press articles from
| Precision | Recall  | F-Measure |
|-----------|---------|-----------|
| 77.98%    | 89.01%  | 83.13%    |

Table 1: *DARE* baseline performance in the Nobel prize domain.

the years 2001 and 2002. From these articles about 37,000 sentences containing at least two *person* named entities are picked. The closed-world knowledge database for this domain consists of the marriage relations of celebrities, i.e. all the (ex-)spouses for 289 famous people are listed, as well as the time span the marriage relation lasted. All of the selected celebrities either appeared on the web page *Celebrity Central*\(^5\) of *People* magazine or where among the *TIME 100* of 2007\(^6\), 2008\(^7\) & 2009\(^8\), a list of the most influential people in the world. The relations were manually gathered from the German and English version of *Wikipedia*. Formally, the closed-world knowledge database is the following set:

\[
cwDB_{celeb} = \{cw_i \mid i \text{ being one of the 289 selected celebrities}\}
\]

where \(cw_i = (W_i, C_i)\) and \(C_i = (\text{NameOfSpouse}, i)\) and \(W_i\) contains the relation instances of all marriages of \(i\).

### 5.2 Nobel Prize Domain

The *DARE* system with the modified ranking is at first run on the Nobel prize corpus. All the experiments in this domain are conducted with the seed: \(\langle \text{Günter Grass}, 1999, \text{ Nobel, literature} \rangle\). Making no special usage of the ranking mechanism described in Section 2.2, the baseline performance of *DARE* is as shown as in Table 1. The precision is calculated by validating the resulting relation instances from the learning process against the data gathered from [http://nobelprize.org](http://nobelprize.org). Note that it is not evaluated whether the sentence, from which a certain instance was extracted, does actually contain a mention of the target relation. Just the truth value of the extracted instance itself is evaluated. The problem of this approach is that a sentence can express a totally different relation than the intended one, and nevertheless contain the named entities of a correct instance of the *Award winning event of Nobel prize laureates* relation, which then results in a false positive. But since a rule matching this dangerous sentence will most likely also match a sentence whose contained named entities are not part of a correct instance of the target relation, the impact of this effect on precision accuracy should be fairly low. The recall value is calculated as the fraction of the extracted

\(^5\)http://www.people.com/people/celebrities
\(^6\)http://www.time.com/time/specials/2007/time100
\(^7\)http://www.time.com/time/specials/2007/0,28757,1733748,00.html
\(^8\)http://www.time.com/time/specials/packages/0,28757,1894410,00.html
| Iter. | # Gen. Rules | # Useless Rules | # Rules w. Sc. 0 | # Rules w. Sc. 1 |
|-------|--------------|----------------|------------------|-----------------|
| 0     | 11           | 2              | 5                | 4               |
| 1     | 816          | 705            | 45               | 66              |
| 2     | 538          | 508            | 12               | 18              |
| 3     | 22           | 21             | 0                | 1               |
| 4     | 2            | 2              | 0                | 0               |
| Sum   | 1389         | 1238           | 62               | 89              |

Table 2: Overview of rule generation by DARE & rule-score distribution of the exclusive ranking with the complete cwDB in the Nobel prize domain. (Iter. is short for Iteration, Gen. Rules for Generated Rules, Useless Rules for Useless Rules, and Rules w. Sc. for Rules with Score.)

| Precision | Recall | F-Measure |
|-----------|--------|-----------|
| 100.00%   | 73.77% | 84.91%    |

Table 3: Performance of the exclusive ranking & complete cwDB in the Nobel prize domain.

correct instances among the contained instances in the corpus. Since using a ranking mechanism, combined with a threshold for rule confidence values, reduces the number of accepted extracted relation instances, recall values stated later in this subsection will all be lower than this one. The F-measure is calculated using the formula: $F$-measure $= 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$.

Table 2 lists the number of rules that were created in each iteration of the learning process. In the first iteration (Iteration 0), the amount of new rules is small because only one instance was used as a seed. These rules extracted instances which turned out be very fruitful seeds for the next iteration, as much more rules were created. After the third iteration (Iteration 2) the learning process is mainly finished. Useless rules do not extract new relation instances. Since the ranking approach of this thesis scores rules according to their extraction performance, useless rules are not scored and are omitted in the figures and tables of this section.

5.2.1 Exclusive Ranking with the Complete cwDB

The main idea of the exclusive ranking is to exclude every rule that extracts at least one relation instance that was identified to be wrong. The remaining rules are utilized for relation extraction. The rule score distribution for this ranking is set out in Table 2. Most of the rules are scored 1, which indicates the good quality of rule creation. In comparison to the baseline performance, using this ranking results in a precision boost from 77.98% to 100%, which is not surprising since the complete cwDB was used. Interestingly, as
Figure 18: Excluded rules with actual good quality. (head denotes the root node of a sentence’s dependency tree or of parts of the tree. object, subject, mod are dependency relations. win is a word stem of a verb (VB). Winner and PrizeName specify the relational argument into which the matching node of a dependency tree is inserted. person and prize are named-entity types that restrict the matching with dependency trees.)

shown in Table 3, the recall decreases more than 15 percentage points compared to the performance of the baseline. A reason for this large recall drop is that even good rules sometimes extract wrong relation instances because of error sources other than the rule itself. These actually good rules are excluded from relation extraction resulting in the mentioned recall drop. Two excluded rules are shown in Figure 18 (in a condensed rule representation). Clearly, these rules do express the target relation and should not be excluded.

One of many error sources is the corpus itself. When corpora are acquired from the Internet using web crawlers, errors in the processing of the document text can occur, resulting in garbled text. Also, there are newspaper articles containing false information, e.g. stating a wrong year or wrong area for a Nobel prize win. The embedment of information in the context of modality or negation is also a possible reason for a good rule to be evaluated as bad. This includes the expression of wishes or speculation by the authors of an article from the corpus. Other error sources are the NER and the dependency parsing applied in the preprocessing of the corpus or an erroneous coreference resolution.

These results show that rule confidence estimation with closed-world knowledge should be done in a softer way. This is realized in the soft ranking.

5.2.2 Soft Ranking

The soft ranking strategy does not exclude any rules, instead it assigns a confidence score to each rule based on the formula in Figure 17. Rules that extract correct instances, more specific relation instances and stem from high-scored seed instances obtain a better score than other patterns. This time, the performance with the complete cwDB as well as with the small-sized cwDB is evaluated.
Figure 19: Correlation of a rule’s extraction precision with its score when using the soft ranking & complete \textit{cwDB} in the Nobel prize domain.

Figure 20: Performance of the soft ranking & complete \textit{cwDB} in the Nobel prize domain.
Figure 21: Distribution of rule scores when using the soft ranking & complete cwDB in the Nobel prize domain.

**Complete cwDB** At first it is examined whether this ranking method shows a good correlation between the extraction precision of rules and their assigned score. Figure 19 shows a graph in which each point stands for a set of rules with the same fraction of correct instances among the extracted ones and with the same assigned score. As expected, using the complete cwDB shows an ideal correlation. Thus there is now the possibility to introduce a threshold for the rule scores to control the minimum extraction precision a rule must have to be accepted. For a given threshold $j$ all rules with $\text{confidence}(\text{rule}) \geq j$ as well as the instances they extract are accepted. Figure 20 depicts the precision, the recall and the F-measure curve for different thresholds. The performance at threshold 0 is the one from Table 1, as all rules are accepted. The higher the threshold is chosen, the higher is the precision, but the lower is the reached recall level. The optimal choice is a threshold of about 0.7 in this case, as this maximizes the F-measure. Figure 21 illustrates the rule-score distribution for this ranking. Only a few rules extract both correct and incorrect instances, namely the ones with a score higher than 0, but lower than 1. In contrast to the exclusive ranking, some of these rules are not excluded, which results in the better recall e.g. at a threshold of 0.7.

**Small-sized cwDB** This experiment investigates the performance of the soft ranking strategy in cases in which only limited knowledge about the target domain is available, i.e. the normal setting of RE. Therefore the results of this experiment give a more realistic analysis of the use of the closed-world ranking approach. The small-sized cwDB
is made up of all Nobel prize winners in the area of peace, which is about one eighth of the size of the complete cwDB. Again, the correlation of the confidence score with a rule’s extraction precision is examined first. This is done in Figure 22. Although the development curve here is not as smooth as that in Figure 19, the higher scored rules have better precision values than most of the lower scored rules. Unfortunately, some good performing rules are assigned a low confidence value, i.e. there are rules in the upper left corner. This indicates that further optimization of the concrete rule scoring function is needed.

The threshold evaluation of Figure 23 shows that the soft ranking can help to improve the performance of RE. With a threshold of 0.4, the system achieves the best F-measure value of this setting. At this point, the precision value is about 11 percentage points above the one from the baseline, while there is only a slight decrease in recall. Compared to the exclusive ranking, the F-measure at this threshold is about 4 points higher, while only a small fraction of the knowledge used in the exclusive-ranking experiment is needed here. These results show that the closed-world ranking strategy indeed helps to improve precision without damaging recall too much.

Figure 24 depicts the distribution of the rule scores. In contrast to Figure 21, the scores are more uniformly distributed. Similar to the rule analysis performed by Xu (2007), a classification of the rules at different thresholds is done, shown in Table 4. The classes were already described in the paragraph *The Problem of Relation Extraction* on page 12. As shown in the table, more than one fourth of the extraction rules created by
Figure 23: Performance of the soft ranking & small-sized $cwDB$ in the Nobel prize domain.

Figure 24: Distribution of rule scores when using the soft ranking & small-sized $cwDB$ in the Nobel prize domain.
### Table 4: Quality analysis of rules at different threshold levels when using the soft ranking & small-sized *cwDB* in the Nobel prize domain. (Threshold abbreviated by *Thresh.*)

| Thresh. | Good     | Dangerous | Bad     |
|---------|----------|-----------|---------|
| Baseline | 58.94% | 26.49%    | 14.57%  |
| 0.1     | 64.96% | 29.20%    | 5.84%   |
| 0.2     | 66.67% | 27.91%    | 5.43%   |
| 0.3     | 69.23% | 26.50%    | 4.27%   |
| **0.4** | **73.27%** | **23.76%** | **2.97%** |
| 0.5     | 76.00% | 22.67%    | 1.33%   |
| 0.6     | 77.59% | 20.69%    | 1.72%   |
| 0.7     | 77.50% | 22.50%    | 0.00%   |
| 0.8     | 87.50% | 12.50%    | 0.00%   |
| 0.9     | 85.71% | 14.29%    | 0.00%   |
| 1.0     | 90.00% | 10.00%    | 0.00%   |

The baseline system are dangerous and almost 15% are plain wrong. Applying the soft ranking with the small-sized *cwDB* increases the fraction of good rules to almost three quarters and nearly eliminates all bad rules at threshold 0.4. By choosing the highest threshold value, even 90% of the rules are good ones, while the remaining rules are at least not totally wrong. This indicates the use of the closed-world ranking, too.

**Comparison to Present DARE Ranking** Section 2.2.3 describes the present ranking mechanism of *DARE*. In contrast to the soft ranking with the small-sized *cwDB*, it uses no additional knowledge for rule confidence estimation, just indicators like the number of sentences a rule matched are exploited. Figure 25 shows the performance of *DARE*’s ranking strategy. Since the ranking function (Figure 12) is not bounded, no thresholds for rule scores are defined. Instead the performance of the $x\%$ highest scored and actively extracting rules is examined, where $x$ starts at 100% (equals baseline performance of Table 1) and is decreased in steps of 5. Doing so produces the curves in Figure 25. Exploiting the rule ranking, i.e. treating rules according to their score, can only slightly improve the precision of the extracted instances, e.g. from almost 80% to nearly 85% precision when using only the 25% best rules. Unfortunately, the price for that increase in precision is an enormous drop in recall. When using the mentioned rule group over 12% of the instances contained in the corpus are lost.

Figure 26 compares the performance of *DARE*’s present ranking with the performance of the soft ranking with the small-sized *cwDB*. In this diagram, the more a curve is located in the upper right, the better its performance. The soft ranking shows a more desirable behavior than the other ranking. On the one hand there is a smaller recall drop that has to be paid to reach a given precision value. On the other hand the present rule ranking is not even capable of reaching 85% precision or more, no matter how much
**Figure 25:** Performance of **DARE**’s present rule ranking in the Nobel prize domain.

**Figure 26:** Performance of the soft ranking & small-sized **cwDB** compared to **DARE**’s present ranking in the Nobel prize domain.
Figure 27: Dangerous rule from meet relation.

| Estimated Precision | # Extracted mentions / instances |
|---------------------|----------------------------------|
| 9.00%               | 25,183                           |

Table 5: DARE baseline performance in the marriages domain.

recall drop is accepted. This comparison therefore proves that the soft ranking approach
is indeed superior to the one followed in the present rule ranking.

5.3 Marriages Domain

As presented in the preceding subsection, the soft ranking strategy delivers a promising
result in the Nobel prize domain. To investigate the usefulness in other domains, another
target relation, marriages, is selected. Because of the large number of possible relations
between two persons, there is the keen risk that Relation Extraction might produce
dangerous or bad rules. For example, the rule shown in Figure 27 is a dangerous one,

| Iter. | # Gen. Rules | # Usel. Rules |
|-------|--------------|---------------|
| 0     | 12           | 7             |
| 1     | 339          | 258           |
| 2     | 8,554        | 7,610         |
| 3     | 3,055        | 2,767         |
| 4     | 275          | 241           |
| 5     | 23           | 23            |

Table 6: Overview of rule generation by DARE in the marriages domain. (Iter. is short for Iteration, Gen. Rules for Generated Rules, Usel. Rules for Useless Rules, and Rules w. Sc. for Rules with Score.)
because it expresses a relation that intersects the target one. Since mentions of *meetings* between married persons (celebrities) in newspapers are often reported, the creation of such a rule during the learning process is highly probable.

For the experiment in this subsection the following two seeds were used: ⟨Lisa Marie Presley, Michael Jackson, 1994⟩ and ⟨Cindy Crawford, Richard Gere, 1991⟩. In this domain, instead of evaluating only the extracted relation instances, the sentence from which an instance was extracted is examined on whether it actually mentions the *marriages* relation or not. Thus, a mentions evaluation is performed. Due to the lack of a gold standard for this domain, only a small amount of samples of the mentions were evaluated by hand to calculate an approximate precision value. Not using any ranking and a sample size of 100 mentions, the baseline performance is as shown in Table 5. Table 6 lists the number of created rules per iteration of the learning cycle and the fraction of them that were useless. Figure 28 shows the rule-score distribution of the useful (non-useless) rules. In contrast to the Nobel prize domain, many rules were low-scored.

Similar to the experiment in the Nobel prize domain, the evaluation is conducted by defining certain thresholds for the rule confidence values and examining the system performance when applying this threshold. For each used threshold 100 sample mentions were manually validated. Table 7 shows the evaluated system performance. Because of the small sample size, the results are not as reliable as the ones from the previous subsection. Since there is no gold standard, the estimated number of correct mentions serves as an approximated recall here. It is calculated by multiplying the precision of

**Figure 28:** Distribution of rule scores when using the soft ranking in the marriages domain.
| Thresh. | # Mentions | Prec. Sam. | Est. # Cor. Men. | # Rules | Good | Dangerous | Bad  |
|---------|------------|------------|-----------------|--------|------|----------|------|
| Baseline | 25,183     | 9.00%      | 2,266           | 1,352  |      |          |      |
| 0.1     | 19,806     | 7.00%      | 1,386           | 562    |      |          |      |
| 0.2     | 14,542     | 9.00%      | 1,309           | 159    |      |          |      |
| 0.3     | 11,259     | 15.00%     | 1,689           | 121    | 19.83% | 33.88%   | 46.28% |
| **0.4** | **788**    | **65.00%** | **512**         | **72** | 25.00% | **27.78%** | **47.22%** |
| 0.5     | 195        | 67.00%     | 131             | 29     | 37.93% | 17.24%   | 44.83% |
| 0.6     | 115        | 84.00%     | 97              | 11     | 45.45% | 27.27%   | 27.27% |
| 0.7     | 55         | 89.09%     | 49              | 6      | 50.00% | 33.33%   | 16.67% |

Table 7: Performance of the soft ranking and a limited *cwDB* in the marriages domain. (*Threshold* abbreviated by *Thresh.*; *Precision of Mention Sample* by *Prec. Sam.*, *Estimated Number of Correct Mentions* by *Est. # Cor. Men.*, respectively.)

```latex
head((marry, VB)),
aux({
    head((be, VB))
}),
dep({
    head([NAMEOFSPouse<person>]),
    dep({
        head([YEAR<year>])
    })
}),
nsubj({
    head([NAMEOFSPouse<person>])
})
```

Figure 29: Good rule for *marriages* relation.
"My kids, I really don't like them to watch that much television," said Cruise, 40, who adopted Isabella and Connor while he was married to second wife Nicole Kidman.

Figure 30: Example sentence from the marriages domain.

Table 7 also contains a quality classification of the learned rules. Because of the large number of mentions, only rules with a score of at least 0.3 were examined. As the table indicates, the usage of the soft ranking does indeed improve the quality of the extraction rules. The manual error analysis showed that wrong coreference resolution is a major error source in this domain. For example, the inability of the NER component to distinguish Prince Charles, the former husband of British princess Diana, from Charles Spencer, her brother, is the reason that DARE shifts to the sibling relation during the learning process. In comparison to the Nobel prize domain, the marriage relation between persons is often used only as additional information to a person which is involved in a reported event. Therefore, correctly resolving coreferences is much more important for the precision of relation extraction in the marriages domain, than it is for the Nobel prize domain. The sentence of Figure 30 contains a typical example for a sentence from the marriages corpus. Wrongly resolving the underlined pronoun with Connor instead of Cruise might lead to the extraction of the wrong instance (Connor, Nicole Kidman, –) by an actually good rule.
6 Conclusion

The goal of this student research project was to develop a way to improve the precision of an existing machine-learning system for relation extraction rules by incorporating previously existing knowledge about a target domain. The proposed approach structures this knowledge in the form of closed worlds, thus allowing to reason about the correctness of some target relation’s instances. Two different strategies for rule confidence estimation using this instance validation were designed. The exclusive ranking strategy turned out to be too strict, i.e. suffering from a large recall drop because of the exclusion of actually good extraction rules. Because of existing inaccuracies in the linguistic preprocessing of the text base, it seemed necessary to soften the impact of instance validation on rule ranking. This resulted in the soft ranking, which combined the closed-world knowledge with the duality principle to estimate the quality of a learned extraction rule. The experiments performed on the Nobel prize domain demonstrated the use of the proposed approach. Even a relatively small amount of domain knowledge, represented by the small-sized cwDB, was sufficient to improve the system performance significantly without hurting the recall too much. An experiment conducted for another target relation, marriages between persons, supports the usefulness of the soft ranking approach.

This work therefore shows that machine-learning systems can take advantage of effective utilization of external knowledge to improve the learning results. At the moment the exploited domain knowledge is realized as a (specially designed) set of sample instances of the target relation. Nevertheless, the proposed ranking strategy is general enough to be extended with different information. This could include e.g. commonsense knowledge about the target relation.
References

Agichtein, E. (2006). Confidence estimation methods for partially supervised information extraction. In Proceedings of the Sixth SIAM International Conference on Data Mining, Bethesda, MD, USA. SIAM.

Agichtein, E. and Gravano, L. (2000). Snowball: extracting relations from large plain-text collections. In DL ’00: Proceedings of the fifth ACM conference on Digital libraries, pages 85–94, New York, NY, USA. ACM.

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

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 1247–1250, New York, NY, USA. ACM.

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

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

Carstensen, K., Ebert, C., Jekat, S., Ebert, C., Langer, H., and Klabunde, R. (2009). Computerlinguistik und Sprachtechnologie: Eine Einführung. Springer.

de Marneffe, M.-C. and Manning, C. D. (2008). Stanford Dependencies manual.

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

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

Fellbaum, C., editor (1998). WordNet: an electronic lexical database. Christiane Fellbaum, Cambridge, MA, USA.

Iacobelli, F., Nichols, N., Birnbaum, L., and Hammond, K. (2010). Finding new information via robust entity detection. In Proactive Assistant Agents (PAA2010) AAAI 2010 Fall Symposium, Arlington, VA, USA.

37
Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, pages 423–430. Association for Computational Linguistics.

Kochhar, S., Mazzocchi, S., and Paritosh, P. (2010). The anatomy of a large-scale human computation engine. In Proceedings of Human Computation Workshop at the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington D.C.

MacEachren, A., Stryker, M., Turton, I., and Pezanowski, S. (2010). Health geojunction: place-time-concept browsing of health publications. International Journal of Health Geographics, 9(1):23.

Nguyen, Q., Tikk, D., and Leser, U. (2010). Simple tricks for improving pattern-based information extraction from the biomedical literature. Journal of Biomedical Semantics, 1(1):9.

Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia. The Association for Computer Linguistics.

Plake, C., Schiemann, T., Pankalla, M., Hakenberg, J., and Leser, U. (2006). Alibaba: Pubmed as a graph. Bioinformatics, 22(19):2444–2445.

Siniakov, P. (2008). GROPUS - an adaptive rule-based algorithm for information extraction. PhD thesis, Freie Universität Berlin.

Stevenson, M. and Greenwood, M. (2005). A semantic approach to IE pattern induction. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05), pages 379–386, Ann Arbor, Michigan. Association for Computational Linguistics.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: A Core of Semantic Knowledge. In 16th international World Wide Web conference (WWW 2007), New York, NY, USA. ACM Press.

Sudo, K., Sekine, S., and Grishman, R. (2003). An improved extraction pattern representation model for automatic IE pattern acquisition. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03), pages 224–231.

Uszkoreit, H., Xu, F., and Li, H. (2009). Analysis and improvement of minimally supervised machine learning for relation extraction. In 14th International Conference on Applications of Natural Language to Information Systems. Springer.
Xu, F. (2007). *Bootstrapping Relation Extraction from Semantic Seeds*. Phd-thesis, Saarland University.

Xu, F., Uszkoreit, H., and Li, H. (2007). A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *Proceedings of ACL 2007, 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic.

Yangarber, R. (2001). *Scenarion Customization for Information Extraction*. Dissertation, Department of Computer Science, Graduate School of Arts and Science, New York University, New York, USA.

Yangarber, R. (2003). Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 343–350, Sapporo Convention Center, Sapporo, Japan.

Yangarber, R., Grishman, R., and Tapanainen, P. (2000). Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 940–946.
Selbstständigkeitserklärung

Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbstständig verfasst wurde und keine außer den angegebenen Quellen verwendet wurden.

Berlin, 28. Oktober 2010

__________________________