A Seed-driven Bottom-up Machine Learning Framework for Extracting Relations of Various Complexity

Feiyu Xu, Hans Uszkoreit and Hong Li
Language Technology Lab, DFKI GmbH
Stuhlsatzenhausweg 3, D-66123 Saarbruecken
{feiyu,uszkoreit,hongli}@dfki.de

Abstract

A minimally supervised machine learning framework is described for extracting relations of various complexity. Bootstrapping starts from a small set of n-ary relation instances as "seeds", in order to automatically learn pattern rules from parsed data, which then can extract new instances of the relation and its projections. We propose a novel rule representation enabling the composition of n-ary relation rules on top of the rules for projections of the relation. The compositional approach to rule construction is supported by a bottom-up pattern extraction method. In comparison to other automatic approaches, our rules cannot only localize relation arguments but also assign their exact target argument roles. The method is evaluated in two tasks: the extraction of Nobel Prize awards and management succession events. Performance for the new Nobel Prize task is strong. For the management succession task the results compare favorably with those of existing pattern acquisition approaches.

1 Introduction

Information extraction (IE) has the task to discover n-tuples of relevant items (entities) belonging to an n-ary relation in natural language documents. One of the central goals of the ACE program[1] is to develop a more systematically grounded approach to IE starting from elementary entities, binary relations to n-ary relations such as events. Current semi- or unsupervised approaches to automatic pattern acquisition are either limited to a certain linguistic representation (e.g., subject-verb-object), or only deal with binary relations, or cannot assign slot filler roles to the extracted arguments, or do not have good selection and filtering methods to handle the large number of tree patterns (Riloff, 1996; Agichtein and Gravano, 2000; Yangarber, 2003; Sudo et al., 2003; Greenwood and Stevenson, 2006; Stevenson and Greenwood, 2006). Most of these approaches do not consider the linguistic interaction between relations and their projections on k dimensional subspaces where $1 \le k < n$, which is important for scalability and reusability of rules. Stevenson and Greenwood (2006) present a systematic investigation of the pattern representation models and point out that substructures of the linguistic representation and the access to the embedded structures are important for obtaining a good coverage of the pattern acquisition. However, all considered representation models (subject-verb-object, chain model, linked chain model and subtree model) are verb-centered. Relations embedded in non-verb constructions such as a compound noun cannot be discovered:

(1) *the 2005 Nobel Peace Prize*

(1) describes a ternary relation referring to three properties of a prize: year, area and prize name. We also observe that the automatically acquired patterns in Riloff (1996), Yangarber (2003), Sudo et al. (2003), Greenwood and Stevenson (2006) cannot be directly used as relation extraction rules because the relation-specific argument role information is missing. E.g., in the management succession domain that concerns the identification of job changing events, a person can either move into a

[1] http://projects.ldc.upenn.edu/ace/

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 584–591,
Prague, Czech Republic, June 2007. ©2007 Association for Computational Linguistics

job (called Person_In) or leave a job (called Person_Out). (2) is a simplified example of patterns extracted by these systems:

(2) *<subject: person> verb <object:organisation>*

In (2), there is no further specification of whether the person entity in the subject position is Person_In or Person_Out.

The ambitious goal of our approach is to provide a general framework for the extraction of relations and events with various complexity. Within this framework, the IE system learns extraction patterns automatically and induces rules of various complexity systematically, starting from sample relation instances as seeds. The arity of the seed determines the complexity of extracted relations. The seed helps us to identify the explicit linguistic expressions containing mentionings of relation instances or instances of their k-ary projections where $1 \leq k < n$. Because our seed samples are not linguistic patterns, the learning system is not restricted to a particular linguistic representation and is therefore suitable for various linguistic analysis methods and representation formats. The pattern discovery is bottom-up and compositional, i.e., complex patterns can build on top of simple patterns for projections.

We propose a rule representation that supports this strategy. Therefore, our learning approach is seed-driven and bottom-up. Here we use dependency trees as input for pattern extraction. We consider only trees or their subtrees containing seed arguments. Therefore, our method is much more efficient than the subtree model of Sudo et al., (2003), where all subtrees containing verbs are taken into account. Our pattern rule ranking and filtering method considers two aspects of a pattern: its domain relevance and the trustworthiness of its origin. We tested our framework in two domains: Nobel Prize awards and management succession. Evaluations have been conducted to investigate the performance with respect to the seed parameters: the number of seeds and the influence of data size and its redundancy property. The whole system has been evaluated for the two domains considering precision and recall. We utilize the evaluation strategy "Ideal Matrix" of Agichtein and Gravano (2000) to deal with unannotated test data.

The remainder of the paper is organised as follows: Section 2 provides an overview of the system architecture. Section 3 discusses the rule representation. In Section 4, a detailed description of the seed-driven bottom-up pattern acquisition is presented. Section 5 describes our experiments with pattern ranking, filtering and rule induction. Section 6 presents the experiments and evaluations for the two application domains. Section 7 provides a conclusion and an outline of future work.

2 System Architecture

Given the framework, our system architecture can be depicted as follows:

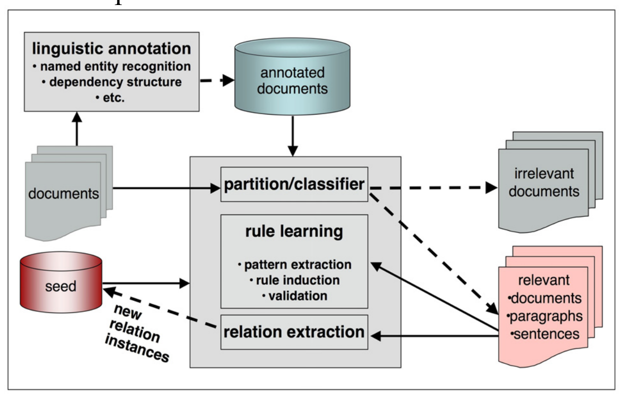

Figure 1. Architecture

This architecture has been inspired by several existing seed-oriented minimally supervised machine learning systems, in particular by Snowball (Agichtein and Gravano, 2000) and ExDisco (Yangarber et al., 2000). We call our system ***DARE***, standing for "***D***omain ***A***daptive ***R***elation ***E***xtraction based on Seeds". ***DARE*** contains four major components: linguistic annotation, classifier, rule learning and relation extraction. The first component only applies once, while the last three components are integrated in a bootstrapping loop. At each iteration, rules will be learned based on the seed and then new relation instances will be extracted by applying the learned rules. The new relation instances are then used as seeds for the next iteration of the learning cycle. The cycle terminates when no new relations can be acquired.

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

The **classifier** has the task to deliver relevant paragraphs and sentences that contain seed elements. It has three subcomponents: document retrieval, paragraph retrieval and sentence retrieval. The document retrieval component utilizes the open source IR-system Lucene[2]. A translation step is built in to convert the seed into the proper IR query format. As explained in Xu et al. (2006), we generate all possible lexical variants of the seed arguments to boost the retrieval coverage and formulate a boolean query where the arguments are connected via conjunction and the lexical variants are associated via disjunction. However, the translation could be modified. The task of paragraph retrieval is to find text snippets from the relevant documents where the seed relation arguments cooccur. Given the paragraphs, a sentence containing at least two arguments of a seed relation will be regarded as relevant.

As mentioned above, the **rule learning** component constitutes the core of our system. It identifies patterns from the annotated documents inducing extraction rules from the patterns, and validates them. In section 4, we will give a detailed explanation of this component.

The **relation extraction** component applies the newly learned rules to the relevant documents and extracts relation instances. The validated relation instances will then be used as new seeds for the next iteration.

3 *DARE* Rule Representation

Our rule representation is designed to specify the location and the role of the arguments w.r.t. the target relation in a linguistic construction. In our framework, the rules should not be restricted to a particular linguistic representation and should be adaptable to various NLP tools on demand. A ***DARE*** rule is allowed to call further ***DARE*** rules that extract a subset of the arguments. Let us step through some example rules for the prize award domain. One of the target relations in the domain is about a person who obtains a special prize in a certain area in a certain year, namely, a quaternary tuple, see (3). (4) is a domain relevant sentence.

(3) <recipient, prize, area, year>

(4) *Mohamed ElBaradei won the <u>2005 Nobel Peace Prize</u> on Friday for his efforts to limit the spread of atomic weapons.*

(5) is a rule that extracts a ternary projection instance <*prize, area, year*> from a noun phrase compound, while (6) is a rule which triggers (5) in its object argument and extracts all four arguments. (5) and (6) are useful rules for extracting arguments from (4).

(5)

Rule name:: prize_area_year_1

Rule:: $(\boxed{3}\ \text{year})(\boxed{1}\ \text{prizename})\ (\boxed{2}\ \text{areaname})\ \text{'Prize'}$

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

(6)

Rule name:: recipient_prize_area_year_1

Rule body::
$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{pos} & \text{verb} \\ \text{mode} & \text{active} \\ \text{wordform} & \text{"win"} \end{bmatrix} \\ \text{subject} & \begin{bmatrix} \text{head} & \boxed{1}\ \text{Person} \\ \text{rule} & \text{recipient\_1::}\ \langle \boxed{1} Person \rangle \end{bmatrix} \\ \text{object} & \begin{bmatrix} \text{head} & \begin{bmatrix} \text{wordform} & \text{"prize"} \end{bmatrix} \\ \text{rule} & \text{prize\_area\_year\_1::}\ \langle \boxed{2} Prize, \boxed{3} Area, \boxed{4} Year \rangle \end{bmatrix} \end{bmatrix}$$

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

Next we provide a definition of a ***DARE*** rule:
A ***DARE*** rule has three components

1. rule name: $\mathbf{r}_i$;
2. output: a set **A** containing the **n** arguments of the n-ary relation, labelled with their argument roles;
3. rule body in AVM format containing:
   - specific linguistic labels or attributes (e.g., **subject**, **object**, **head**, **mod**), derived from the linguistic analysis, e.g., dependency structures and the named entity information
   - **rule**: its value is a ***DARE*** rule which extracts a subset of arguments of **A**

The rule in (6) is a typical ***DARE*** rule. Its subject and object descriptions call appropriate ***DARE*** rules that extract a subset of the output relation arguments. The advantages of this rule representation strategy are that (1) it supports the bottom-up rule composition; (2) it is expressive enough for the representation of rules of various complexity; (3) it reflects the precise linguistic relationship among the relation arguments and reduces the template merging task in the later phase; (4) the rules for the subset of arguments may be reused for other relation extraction tasks.

The rule representation models for automatic or unsupervised pattern rule extraction discussed by

[2] http://www.lucene.de

Stevenson and Greenwood (2006) do not account for these considerations.

4 Seed-driven Bottom-up Rule Learning

Two main approaches to seed construction have been discussed in the literature: pattern-oriented (e.g., ExDisco) and semantics-oriented (e.g., Snowball) strategies. The pattern-oriented method suffers from poor coverage because it makes the IE task too dependent on one linguistic representation construction (e.g., *subject-verb-object*) and has moreover ignored the fact that semantic relations and events could be dispersed over different sub-structures of the linguistic representation. In practice, several tuples extracted by different patterns can contribute to one complex relation instance.

The semantics-oriented method uses relation instances as seeds. It can easily be adapted to all relation/event instances. The complexity of the target relation is not restricted by the expressiveness of the seed pattern representation. In Brin (1998) and Agichtein and Gravano (2000), the semantics-oriented methods have proved to be effective in learning patterns for some general binary relations such as *booktitle-author* and *company-headquarter* relations. In Xu et al. (2006), the authors show that at least for the investigated task it is more effective to start with the most complex relation instance, namely, with an n-ary sample for the target n-ary relation as seed, because the seed arguments are often centred in a relevant textual snippet where the relation is mentioned. Given the bottom-up extracted patterns, the task of the rule induction is to cluster and generalize the patterns. In comparison to the bottom-up rule induction strategy (Califf and Mooney, 2004), our method works also in a compositional way. For reasons of space this part of the work will be reported in Xu and Uszkoreit (forthcoming).

4.1 Pattern Extraction

Pattern extraction in ***DARE*** aims to find linguistic patterns which do not only trigger the relations but also locate the relation arguments. In ***DARE***, the patterns can be extracted from a phrase, a clause or a sentence, depending on the location and the distribution of the seed relation arguments.

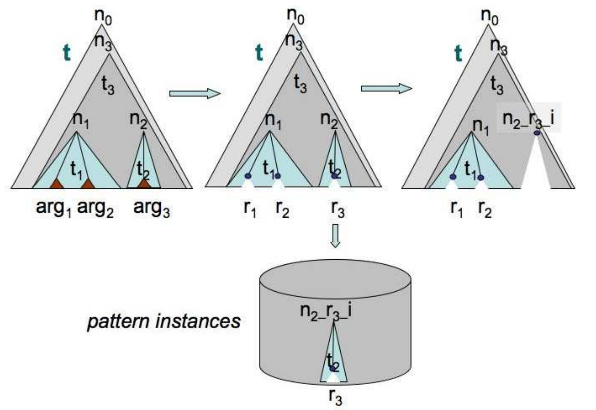

Figure 2. Pattern extraction step 1

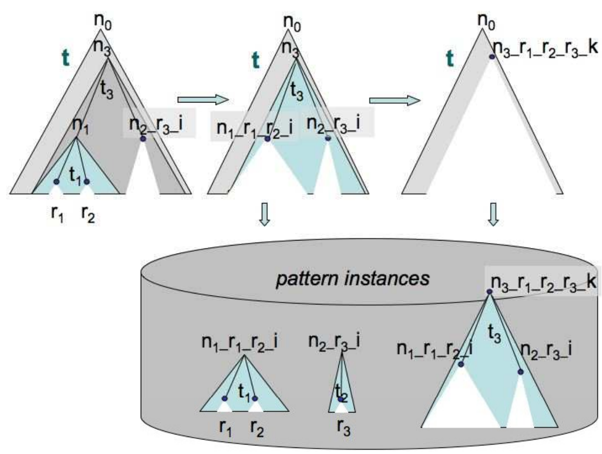

Figure 3. Pattern extraction step 2

Figures 2 and 3 depict the general steps of bottom-up pattern extraction from a dependency tree $\boldsymbol{t}$ where three seed arguments $\boldsymbol{arg}_1$, $\boldsymbol{arg}_2$ and $\boldsymbol{arg}_3$ are located. All arguments are assigned their relation roles $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ and $\boldsymbol{r}_3$. The pattern-relevant subtrees are trees in which seed arguments are embedded: $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ and $\boldsymbol{t}_3$. Their root nodes are $\boldsymbol{n}_1$, $\boldsymbol{n}_2$ and $\boldsymbol{n}_3$. Figure 2 shows the extraction of a unary pattern $\boldsymbol{n}_2\_\boldsymbol{r}_3\_\boldsymbol{i}$, while Figure 3 illustrates the further extraction and construction of a binary pattern $\boldsymbol{n}_1\_\boldsymbol{r}_1\_\boldsymbol{r}_2\_\boldsymbol{j}$ and a ternary pattern $\boldsymbol{n}_3\_\boldsymbol{r}_1\_\boldsymbol{r}_2\_\boldsymbol{r}_3\_\boldsymbol{k}$. In practice, not all branches in the subtrees will be kept. In the following, we give a general definition of our seed-driven bottom-up pattern extraction algorithm:

input: (i) relation = <$r_1, r_2, ..., r_n$>: the target relation tuple with n argument roles.
T: a set of linguistic analysis trees annotated with i seed relation arguments ($1 \leq i \leq n$)

output: P: a set of pattern instances which can extract i or a subset of i arguments.

Pattern extraction:
for each tree $t \in T$

Step 1: (depicted in Figure 2)

1. replace all terminal nodes that are instantiated with the seed arguments by new nodes. Label these new nodes with the seed argument roles and possibly the corresponding entity classes;
2. identify the set of the lowest nonterminal nodes N_1 in t that dominate only one argument (possibly among other nodes).
3. substitute N_1 by nodes labelled with the seed argument roles and their entity classes
4. prune the subtrees dominated by N_1 from t and add these subtrees into P. These subtrees are assigned the argument role information and a unique id.

Step2: For i=2 to n: (depicted in Figure 3)

1. find the set of the lowest nodes N_1 in t that dominate in addition to other children only i seed arguments;
2. substitute N_1 by nodes labelled with the i seed argument role combination information (e.g., $r_i\_r_j$) and with a unique id.
3. prune the subtrees T_i dominated by N_i in t;
4. add T_i to P together with the argument role combination information and the unique id

With this approach, we can learn rules like (6) in a straightforward way.

4.2 Rule Validation: Ranking and Filtering

Our ranking strategy has incorporated the ideas proposed by Riloff (1996), Agichtein and Gravano (2000), Yangarber (2003) and Sudo et al. (2003). We take two properties of a pattern into account:

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

In Riloff (1996) and Sudo et al. (2003), the relevance of a pattern is mainly dependent on its occurrences in the relevant documents vs. the whole corpus. Relevant patterns with lower frequencies cannot float to the top. It is known that some complex patterns are relevant even if they have low occurrence rates. We propose a new method for calculating the domain relevance of a pattern. We assume that the domain relevance of a pattern is dependent on the relevance of the lexical terms (words or collocations) constructing the pattern, e.g., the domain relevance of (5) and (6) are dependent on the terms "prize" and "win" respectively. Given *n* different domains, the domain relevance score (*DR*) of a term *t* in a domain d_i is:

$$DR(t, d_i) = \begin{cases} 0, \text{ if } df(t, d_i) = 0; \\ \dfrac{df(t,d_i)}{N \times D} \times LOG\left(n \times \dfrac{df(t,d_i)}{\sum_{j=1}^{n} df(t,d_j)}\right), \text{ otherwise} \end{cases}$$

where

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

Here the domain relevance of a term is dependent both on its document frequency and its document frequency distribution in other domains. Terms mentioned by more documents within the domain than outside are more relevant (Xu et al., 2002). In the case of $n=3$ such different domains might be, e.g., *management succession*, *book review* or *biomedical texts*. Every domain corpus should ideally have the same number of documents and similar average document size. In the calculation of the trustworthiness of the origin, we follow Agichtein and Gravano (2000) and Yangarber (2003). Thus, the relevance of a pattern is dependent on the relevance of its terms and the score value of the most trustworthy seed from which it origins. Finally, the score of a pattern ***p*** is calculated as follows:

$$score(\boldsymbol{p}) = \sum_{i=0}^{|T|} DR(t_i) \times \max\{score(s) : s \in Seeds\}$$

where $|T| > 0$ and $t_i \in T$

- *T: is the set of the terms occur in p;*
- *Seeds: a set of seeds from which the pattern is extracted;*
- *score(s): is the score of the seed s;*

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

5 Top down Rule Application

After the acquisition of pattern rules, the ***DARE*** system applies these rules to the linguistically annotated corpus. The rule selection strategy moves from complex to simple. It first matches the most complex pattern to the analyzed sentence in order to extract the maximal number of relation arguments. According to the duality principle (Yangarber 2001), the score of the new extracted relation instance *S* is dependent on the patterns from which it origins. Our score method is a simplified version of that defined by Agichtein and Gravano (2000):

$$\mathrm{score}(S)=1-\prod_{i=0}^{|P|}(1-score(P_i))$$

where $P=\{P_i\}$ is the set of patterns that extract S.

The extracted instances can be used as potential seeds for the further pattern extraction iteration, when their scores are validated. The initial seeds obtain 1 as their score.

6 Experiments and Evaluation

We apply our framework to two application domains: Nobel Prize awards and management succession events. Table 1 gives an overview of our test data sets.

Data Set Name	Doc Number	Data Amount
Nobel Prize A (1999-2005)	2296	12,6 MB
Nobel Prize B (1981-1998)	1032	5,8 MB
MUC-6	199	1 MB

Table1. Overview of Test Data Sets.

For the Nobel Prize award scenario, we use two test data sets with different sizes: Nobel Prize A and Nobel Prize B. They are Nobel Prize related articles from New York Times, online BBC and CNN news reports. The target relation for the experiment is a quaternary relation as mentioned in (3), repeated here again:

<recipient, prize, area, year>

Our test data is not annotated with target relation instances. However, the entire list of Nobel Prize award events is available for the evaluation from the Nobel Prize official website[3]. We use it as our reference relation database for building our Ideal table (Agichtein and Gravano, 2000).

For the management succession scenario, we use the test data from MUC-6 (MUC-6, 1995) and define a simpler relation structure than the MUC-6 scenario template with four arguments:

<Person_In, Person_Out, Position, Organisation>

In the following tables, we use PI for Person_In, PO for Person_Out, POS for Position and ORG for Organisation. In our experiments, we attempt to investigate the influence of the size of the seed and the size of the test data on the performance. All these documents are processed by named entity recognition (Drozdzynski et al., 2004) and dependency parser MINIPAR (Lin, 1998).

6.1 Nobel Prize Domain Evaluation

For this domain, three test runs have been evaluated, initialized by one randomly selected relation instance as seed each time. In the first run, we use the largest test data set Nobel Prize A. In the second and third runs, we have compared two random selected seed samples with 50% of the data each, namely Nobel Prize B. For data sets in this domain, we are faced with an evaluation challenge pointed out by DIPRE (Brin, 1998) and Snowball (Agichtein and Gravano, 2000), because there is no gold-standard evaluation corpus available. We have adapted the evaluation method suggested by Agichtein and Gravano, i.e., our system is successful if we capture one mentioning of a Nobel Prize winner event through one instance of the relation tuple or its projections. We constructed two tables (named Ideal) reflecting an approximation of the maximal detectable relation instances: one for Nobel Prize A and another for Nobel Prize B. The Ideal tables contain the Nobel Prize winners that co-occur with the word “Nobel” in the test corpus. Then precision is the correctness of the extracted relation instances, while recall is the coverage of the extracted tuples that match with the Ideal table. In Table 2 we show the precision and recall of the three runs and their random seed sample:

Data Set	Seed	Precision	Recall	
			total	time interval
Nobel Prize A	[Zewail, Ahmed H], nobel, chemistry,1999	**71,6%**	**50,7%**	**70,9%** (1999-2005)
Nobel Prize B	[Sen, Amartya], nobel, economics, 1998	**87,3%**	**31%**	**43%** (1981-1998)
Nobel Prize B	[Arias, Oscar], nobel, peace, 1987	**83,8%**	**32%**	**45%** (1981-1998)

Table 2. Precision, Recall against the Ideal Table

The first experiment with the full test data has achieved much higher recall than the two experiments with the set Nobel Prize B. The two experiments with the Nobel Prize B corpus show similar

3 http://nobelprize.org/

performance. All three experiments have better recalls when taking only the relation instances during the report years into account, because there are more mentionings during these years in the corpus. Figure (6) depicts the pattern learning and new seed extracting behavior during the iterations for the first experiment. Similar behaviours are observed in the other two experiments.

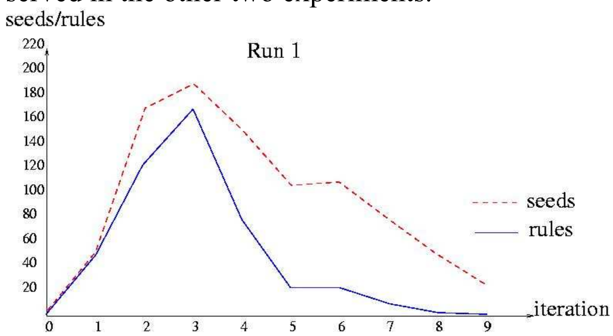

Figure 6. Experiment with Nobel Prize A

6.2 Management Succession Domain

The MUC-6 corpus is much smaller than the Nobel Prize corpus. Since the gold standard of the target relations is available, we use the standard IE precision and recall method. The total gold standard table contains 256 event instances, from which we randomly select seeds for our experiments. Table 3 gives an overview of performance of the experiments. Our tests vary between one seed, 20 seeds and 55 seeds.

Initial Seed Nr.		Precision	Recall
1	A	12.6%	7.0%
	B	15.1%	21.8%
20		48.4%	34.2%
55		62.0%	48.0%

Table 3. Results for various initial seed sets

The first two one-seed tests achieved poor performance. With 55 seeds, we can extract additional 67 instances to obtain the half size of the instances occurring in the corpus. Table 4 show evaluations of the single arguments. B works a little better because the randomly selected single seed appears a better sample for finding the pattern for extracting PI argument.

Arg	precision (A)	precision (B)	Recall (A)	Recall (B)
PI	10.9%	15.1%	8.6%	34.4%
PO	28.6%	–	2.3%	2.3%
ORG	25.6%	100%	2.6%	2.6%
POS	11.2%	11.2%	5.5%	5.5%

Table 4. Evaluation of one-seed tests (A and B)

Table 5 shows the performance with 20 and 55 seeds respectively. Both of them are better than the one-seed tests, while 55 seeds deliver the best performance in average, in particular, the recall value.

arg	precision (20)	precision (55)	recall (20)	recall (55)
PI	84%	62.8%	27.9%	56.1%
PO	41.2%	59%	34.2%	31.2%
ORG	82.4%	58.2%	7.4%	20.2%
POS	42%	64.8%	25.6%	30.6%

Table 5. Evaluation of 20 and 55 seeds tests

Our result with 20 seeds (precision of 48.4% and recall of 34.2%) is comparable with the best result reported by Greenwood and Stevenson (2006) with the linked chain model (precision of 0.434 and recall of 0.265). Since the latter model uses patterns as seeds, applying a similarity measure for pattern ranking, a fair comparison is not possible. Our result is not restricted to binary relations and our model also assigns the exact argument role to the Person role, i.e. Person_In or Person_Out.

We have also evaluated the top 100 event-independent binary relations such as Person-Organisation and Position-Organisation. The precision of these by-product relations of our IE system is above 98%.

7 Conclusion and Future Work

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

The success of the bootstrapping approach crucially depends on the nature of the training data base. One of the most relevant properties of this data base is the ratio of documents to relation instances. Several independent reports of an instance usually yield a higher number of patterns.

The two tasks we used to investigate our method drastically differ in this respect. The Nobel Prize domain we selected as a learning domain for general award events since it exhibits a high degree of redundancy in reporting. A Nobel Prize triggers more news reports than most other prizes. The achieved results met our expectations. With one randomly selected seed, we could finally extract most relevant events in some covered time interval.

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

The conclusion, we draw from the observed difference between the two tasks is simple: We shall always try to find a highly redundant training data set. If at all possible, the training data should exhibit a skewed distribution of reports to events. Actually, such training data may be the only realistic chance for reaching a large number of rare patterns. In future work we will try to exploit the web as training resource for acquiring patterns while using the parsed domain data as the source for obtaining new seeds in bootstrapping the rules before applying these to any other nonredundant document base. This is possible because our seed tuples can be translated into simple IR queries and further linguistic processing is limited to the retrieved candidate documents.

Acknowledgement

The presented research was partially supported by a grant from the German Federal Ministry of Education and Research to the project Hylap (FKZ: 01IWF02) and EU–funding for the project RASCALLI. Our special thanks go to Doug Appelt and an anonymous reviewer for their thorough and highly valuable comments.

References

E. Agichtein and L. Gravano. 2000. *Snowball: extracting relations from large plain-text collections.* In ACM 2000, pages 85–94, Texas, USA.

S. Brin. *Extracting patterns and relations from the World-Wide Web*. In Proc. 1998 Int'l Workshop on the Web and Databases (WebDB '98), March 1998.

M. E. Califf and R. J. Mooney. 2004. Bottom-Up Relational Learning of Pattern Matching Rules for Information Extraction. *Journal of Machine Learning Research*, MIT Press.

W. Drozdzynski, H.-U.Krieger, J. Piskorski; U. Schäfer, and F. Xu. 2004. Shallow Processing with Unification and Typed Feature Structures — Foundations and Applications. *Künstliche Intelligenz* 1:17—23.

M. A. Greenwood and M. Stevenson. 2006. *Improving Semi-supervised Acquisition of Relation Extraction Patterns*. In Proc. of the Workshop on Information Extraction Beyond the Document, Australia.

D. Lin. 1998. Dependency-based evaluation of MINIPAR. In Workshop on the Evaluation of Parsing Systems, Granada, Spain.

MUC. 1995. Proceedings of the Sixth Message Understanding Conference (MUC-6), Morgan Kaufmann.

E. Riloff. 1996. *Automatically Generating Extraction Patterns from Untagged Text.* In Proc. of the Thirteenth National Conference on Articial Intelligence, pages 1044–1049, Portland, OR, August.

M. Stevenson and Mark A. Greenwood. 2006. *Comparing Information Extraction Pattern Models*. In Proc. of the Workshop on Information Extraction Beyond the Document, Sydney, Australia.

K. Sudo, S. Sekine, and R. Grishman. 2003. *An Improved Extraction Pattern Representation Model for Automatic IE Pattern Acquisition*. In Proc. of ACL-03, pages 224–231, Sapporo, Japan.

R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. 2000. *Automatic Acquisition of Domain Knowledge for Information Extraction*. In Proc. of COLING 2000, Saarbrücken, Germany.

R. Yangarber. 2003. *Counter-training in the Discovery of Semantic Patterns.* In Proceedings of ACL-03, pages 343–350, Sapporo, Japan.

F. Xu, D. Kurz, J. Piskorski and S. Schmeier. 2002. *A Domain Adaptive Approach to Automatic Acquisition of Domain Relevant Terms and their Relations with Bootstrapping*. In Proc. of LREC 2002, May 2002.

F. Xu, H. Uszkoreit and H. Li. 2006. *Automatic Event and Relation Detection with Seeds of Varying Complexity*. In Proceedings of AAAI 2006 Workshop Event Extraction and Synthesis, Boston, July, 2006.