# Analysis and Improvement of Minimally Supervised Machine Learning for Relation Extraction

Hans Uszkoreit and Feiyu Xu and Hong Li

DFKI GmbH, LT Lab
Stuhlsatzenhausweg 3, D-66123 Saarbrücken
{uszkoreit,feiyu,lihong}@dfki.de

**Abstract.** The main contribution of this paper is a systematic analysis of a minimally supervised machine learning method for relation extraction grammars. The method is based on a bootstrapping approach in which the bootstrapping is triggered by semantic seeds. The starting point of our analysis is the pattern-learning graph which is a subgraph of the bipartite graph representing all connections between linguistic patterns and relation instances exhibited by the data. It is shown that the performance of such general learning framework for actual tasks is dependent on certain properties of the data and on the selection of seeds. Several experiments have been conducted to gain explanatory insights into the interaction of these two factors. From the investigation of more effective seeds and benevolent data we understand how to improve the learning in less fortunate configurations. A relation extraction method only based on positive examples cannot avoid all false positives, especially when the data properties yield a high recall. Therefore, negative seeds are employed to learn negative patterns, which boost precision.

## 1 Introduction

The charm and the power of the seed-based minimally supervised machine learning method within a bootstrapping framework has been widely recognized and frequently employed for various information extraction tasks (e.g., [8, 10, 6, 3, 12, 7, 4]). The approach has evolved into an empirically promising and theoretically attractive research strand, dedicated to the automatic acquisition of extraction patterns or rules from unannotated textual data (e.g., [18, 6, 5, 3, 1, 2, 9, 15, 14, 16, 13] ). The only task-specific knowledge provided to the automatic learning process is a small set of examples of either patterns or semantic instances. Several methods have been developed that accomplish rather decent results with a minimum of effort [6, 3, 12, 9, 11]. It is mentioned in all these approaches that their seed selection is random, without any explicit criteria. In fact, some experiments conducted by [9] for named entity extraction and those by [13] for relation extraction demonstrate that certain data properties yield a high recall even with a very small number of seed examples.

[13] investigates the role of the seed selection in connection with the data properties in a careful way. Through some dedicated experiments we could obtain

and confirm some new insights on the relevant properties of seed and data. From these insights we derive proposed solutions, some of which could already be empirically validated. All measures for improving recall at the same time trigger false positives. We have observed various sources of degrading precision and propose to employ negative patterns as filters for some classes of false positives. In order to learn such negative patterns we need negative seeds. We will explain the construction of negative seeds and discuss some problems for generalizing these to effective negative rules.

The remainder of the paper is organized as follows: Section 2 describes the state of the art of approaches to seed construction and provides a systematic analysis of the properties of the seed construction and provides the first solutions to lucky seed. Section 3 shows the interaction between the complexity of the target relation, the seed construction and the performance. Section 4 presents the experimental results of the integration of negative rules. Section 5 concludes and opens discussions for the future research.

## 2 Magic and Challenges around Seed

### 2.1 Seed Representation

With respect to the employed seeds, the bootstrapping approaches to relation extraction fall in two classes:

- pattern based seeds
- semantics (relation instance) based seeds

The former class uses simple linguistic patterns that indicate the target relation (e.g., [17, 12, 7]). A typical example of this tradition is the ExDisco system [17]. (1) is an example pattern of the management succession domain, e.g.,

**(1)** *subj(company) verb("appoint") obj(person)*

Patterns serve as structured queries for document retrieval (e.g., [12, 7]). If new patterns from the retrieved documents are found, they will be used to retrieve more documents again. Each iteration in the boostrapping lerning process contains two steps: patterns ($p_i$) extraction and documents ($m_n$) retrieval as depicted in Figure 1.

In general, these patterns cannot be applied for relation extraction straightforwardly because there is no semantic role labelling information between the linguistic arguments in the patterns and semantic arguments specified in the target relations. In example (1) mentioned above, the object argument does not contain the information that it fills the role for a new person which takes over the position. Furthermore, the linguistic patterns are mostly flatter lists of grammatical functions such as a "subj-verb-obj" sequence and do not allow or even afford recursive or hierarchical structures. Therefore, these pattern-based approaches are especially suitable for simple relations that can be expressed by such simple

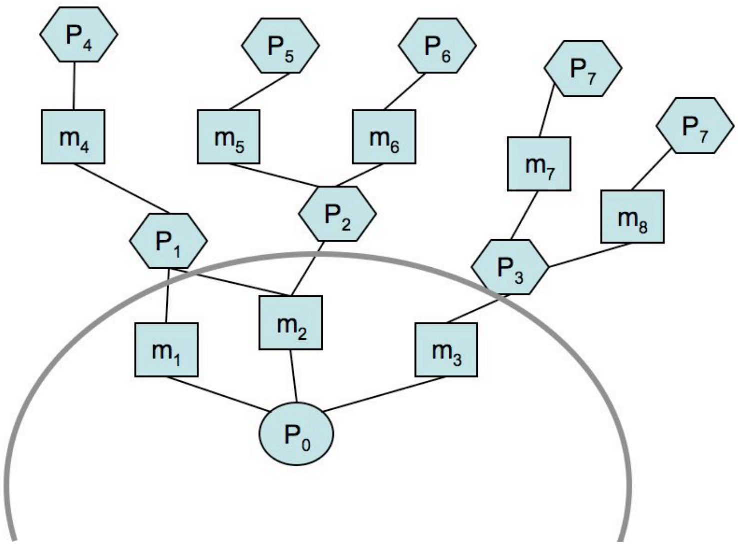



Fig. 1.: Learning graph starting from pattern-based seed. $p_i$: patterns, $m_j$: textual snippets/documents

linguistic patterns. However, complex relations with more arguments often cannot be expressed by such simple patterns. Let us consider an example relation from the prize award domain. The relation contains four arguments representing an event in which a person or an organization won a particular prize in a specific area and in a certain year:

**(2)** *<recipient, award, area, year>*

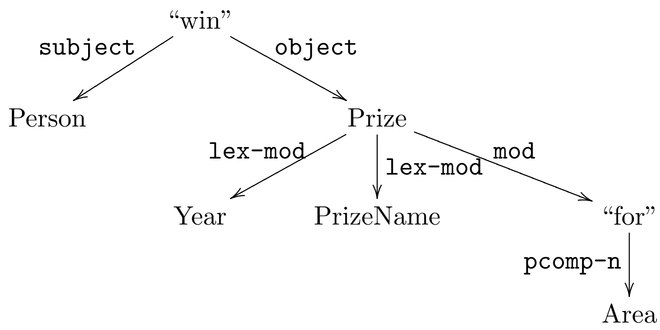
(3) is an example relation instance of (2), referring to an event mentioned in the sentence (4).

**(3)** *<Mohamed ElBaradei, Nobel, Peace, 2005>*

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

(5) is a simplified dependency tree of the parsing result of (4). The pattern-based approaches mentioned above do not provide strategies to extract hierarchical patterns like (5).

**(5)**




Moreover, it is difficult for domain experts but non-linguists to formulate a pattern like (5). In comparison to pattern-based approaches a seed example in the semantics-based approach is very easy to formulate, even for complex relations. It is simply an instance of the target relation, namely, a database record like (3). Users just have to provide some examples and no linguistic knowledge is needed.

In comparison to the pattern-based approach, each iteration in the learning process of the semantics based approach contains one more step, namely, the instance ($e_i$) extraction (e.g., [6, 3]), see Figure 2. In Figure 2, the learning process starts with the instances (e.g., $e_1$) as seed and find textual snippets (e.g., $m_1$, $m_2$, $m_3$) which match the seed and then extract pattern rules (e.g, $r_1$, $r_2$, $r_3$). The learned pattern rules can be applied to the documents to discover new instances. The new instances can be used as seed for the next iteration again. Hence, a kind of co-training takes place here: two classifiers are applied to the textual snippets ($m_i$) to decide whether they mention the target relation (see [5]):

- one utilizes the semantic-instance as feature and
- another relies on the pattern rules.

Among the approaches in this class, [13] and [16] present a general framework for extracting relations of various complexity, called DARE (Domain Adaptive Relation Extraction), starting from a small set of n-ary relation instances as "seed". DARE presents a novel rule representation model which enables the composition of n-ary relation rules on top of the rules for projections of the relation. The compositional approach to rule construction is supported by a bottom-up pattern extraction method. Thus, DARE is able to extract and build hierarchical patterns like (5). DARE learns three rules from the tree in (5), i.e., (6), (7) and (8). (6) and (7) are projection rules, while (8) can cover all four arguments and is equal to the hierarchical pattern (5).

(6) extracts the semantic argument area from the prepositional phrase headed by the preposition "for", while (7) extracts the three arguments *year*, *prize* and *area* from the complex noun phrase and calls the rule (6) for the semantic argument area.

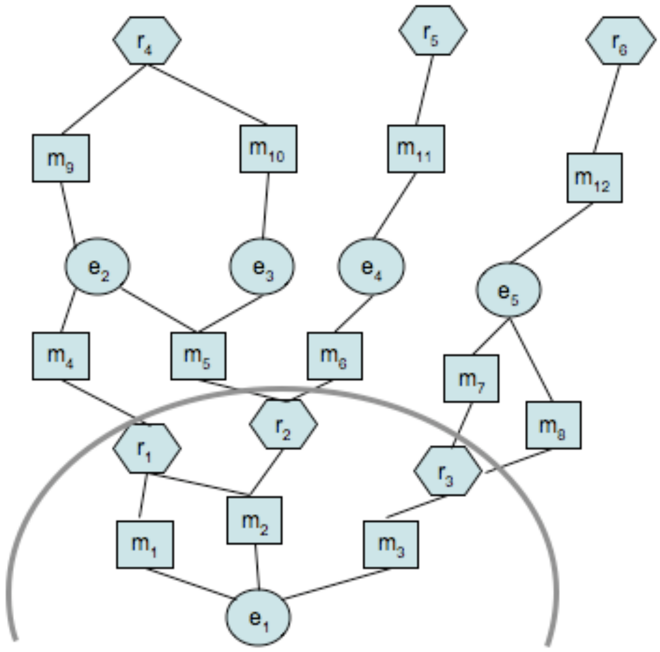



Fig. 2.: Learning graph starting from semantics-based seed. $e_i$: relation instances; $r_i$: extraction rules, mj: textual snippets

**(6)** Rule name :: area_1

$$\text{Rule body} :: \left[\begin{array}{ll} \text{head} & \left[\begin{array}{ll} \text{pos} & \text{noun} \\ \text{lex-form} & \text{“for”} \end{array}\right] \\ & \\ \text{daughters} & <\left[\text{pcomp-n} \quad \left[\text{head} \quad \boxed{1}\ \text{Area}\right]\right]> \end{array}\right]$$

$$\text{Output} :: < \boxed{1} Area >$$

**(7)** Rule name :: year_prize_area_1

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

Output :: < $\boxed{1}$*Year*, $\boxed{2}$*Prize*, $\boxed{3}$*Area* >

(8) is the rule that extracts all four arguments from the verb phrase dominated by the verb “win” and calls (7) to handle the arguments embedded in the linguistic argument “object”.

**(8)** Rule name :: recipient_prize_area_year_1
Rule body ::

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

Output :: < $\boxed{1}$*Recipient*, $\boxed{2}$*Prize*, $\boxed{3}$*Area*, $\boxed{4}$*Year* >

A freely accessible online demonstration `http://dare.dfki.de` illustrates the learning process of DARE for the prize award domain. As demonstrated through the examples of DARE rules, an important advantage of this approach is that the learned rules do not only represent the linguistic patterns but also contain the relevant semantic role labels. Thus, rules learned by this strategy are real extraction rules.

### 2.2 Seed Magic and Lucky Seed

Although the seed plays a central role in the minimally supervised machine-learning framework, its construction is often left underspecified in the literature. Seeds are either randomly chosen or they are stipulated by users (e.g., [3, 16]).

As illustrated in Figure 3, a bipartite graph can describe the connections between all mentions of instances and the patterns (or rules) that express these instances in the learning data corpus. The correct part of the learning graph is a subgraph of the bipartite graph containing all instances recognized and all the rules that were correctly constructed. If the learning starts with several instance examples, the learning graph can be composed of several subgraphs. Some of these may be small islands containing very few connections, while others appear as continents with many connections. In Figure 3, the learning graph contains a big continent and two small islands.

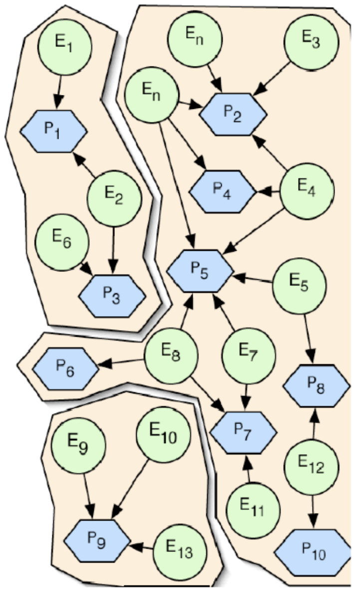
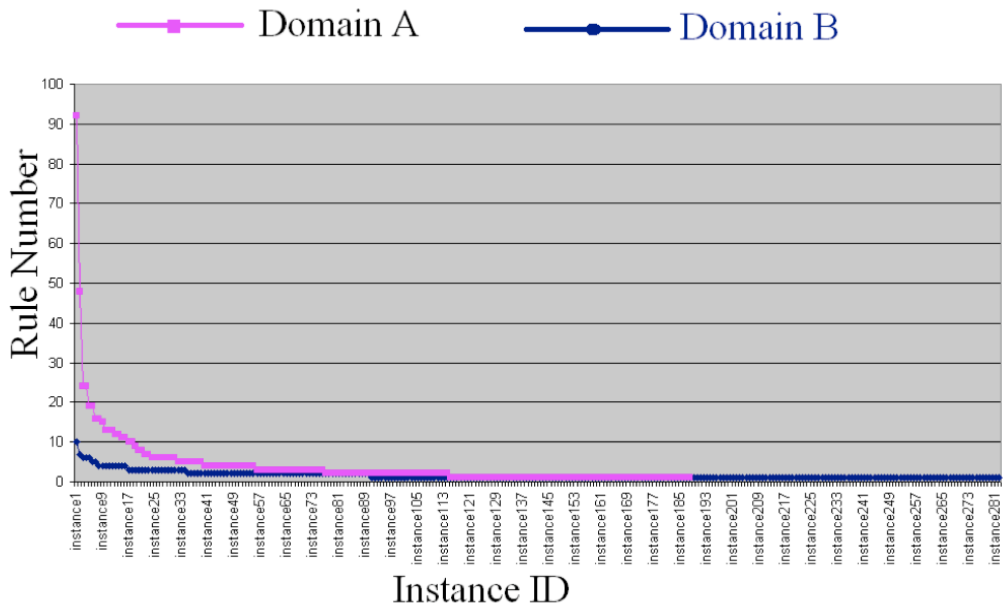



Fig. 3.: Bipartie Graph of Interplay between Instances and Rules

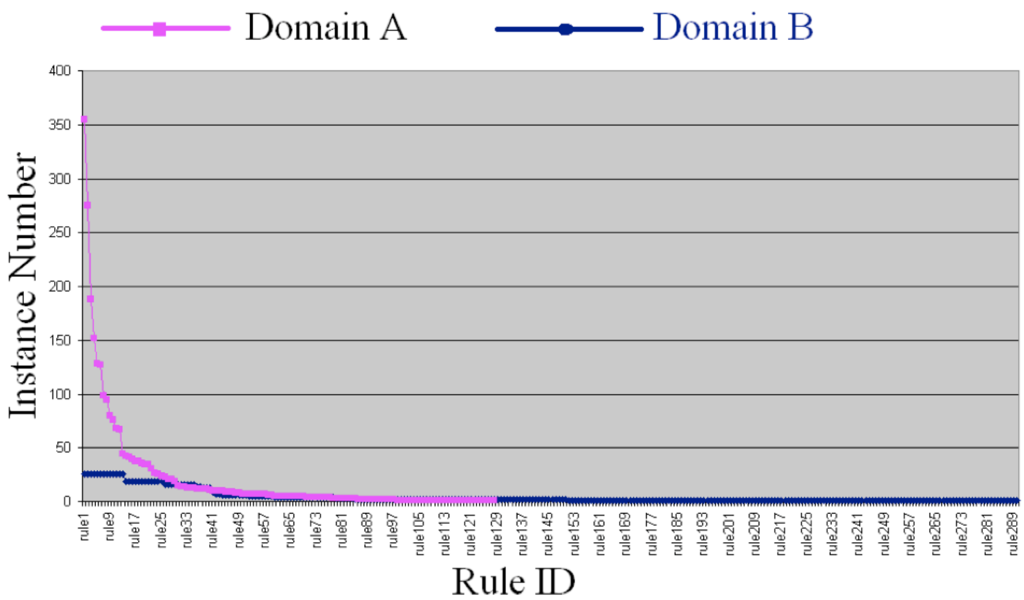
[13] reported on two experiments with two different domains A and B each with a distinct corpus. Corpus A contains almost 2300 documents and is more than ten size larger than corpus B. Corpus A is collected from various newspapers and corpus B comes from one newspaper. The target relations for both domains are quaternary relations.

Figure 4a and Figure 4b show the distribution of the connectivity degree between learned rules and instances in the two domains.

In Domain A, one semantic-seed is sufficient to find a large number of rules (Figure 4a) and one rule suffices to extract many instances (Figure 4b). The distribution behavior in Domain A corresponds to Zipfs law and confirms the results reported by [9], namely, the Power Law distribution. Both distributions are skewed, but in the domain A, we get some sort of scale-free graph, i.e., $P(k){\sim}k^r$. In this case, the graph displays the so-called small world property. We can reach nearly all events from any seed in a few steps, even if the graph grows. The reason is simple: in order to learn the numerous less frequent patterns in the heavy tail of the distribution, we need “event hubs”. But we need the less frequent patterns in order to get to many events mentioned only once.




(a) Instance to Rule Connections




(b) Rule to Instance Relation

Fig. 4.: distribution of the connectivity degree between learned rules and instances in domains

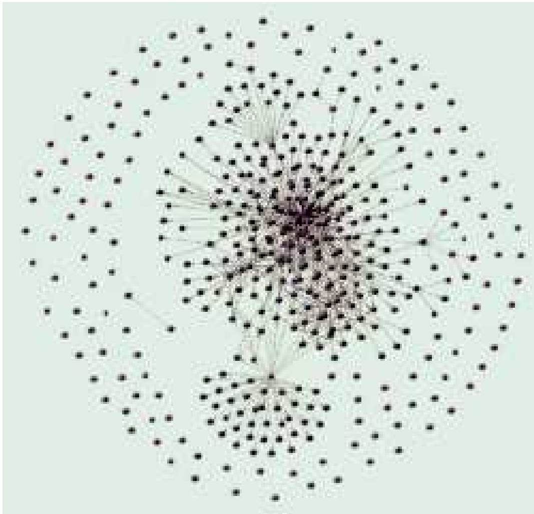


Fig. 5.: Connectivity of Instances in Domain A

The connected instances in domain A in the bootstrapping process can be visualized as a graph with one large component (continent) and many small ones (islands) (see Figure 5). In Figure 5, there are 138 components in the graph. The largest graph contains 298 mentioned instances, while one island contains 2 instances and 136 islands only have one instance. In order to have 100% recall, we would need 138 seed examples for Domain A corpus, of which 137 would have to miraculously match the instances on the islands. However, one seed example in the biggest component can already achieve a maximal recall of 68%.

Thus any instance on the continent is a lucky seed. Such seed triggers the apparent magic reported in the literature: for domain A, a precision of 80.59% and a recall of 62.9% could be achieved with only one example as seed.

The connectivity behavior in Domain B is completely different from Domain A. We find a boring distribution: most instances are just mentioned a single time. Patterns and instances have a very small degree of connectivity (Figure 4a and Figure 4b). Thus, more instances are needed as seed to discover enough patterns. Even with 20 seed examples, its precision is 48.4% and recall 34.2%. 55 examples yield a precision of 62.0% and a recall of 48.0%. The chances for a lucky seed are rather small for such corpus. Thus the magic fails in this case.

## 3 Seed Properties

In the last section, we discussed the advantages of semantic seeds and also the relationship between properties of the data and lucky seeds. However, a pattern seed has more descriptive content than a semantic seed. A semantic seed is simply a list of semantic arguments. As we know, the same combination of semantic arguments can also occur in mentions of different relations. The simpler the combination, the more likely is its occurrence in mentions of multiple relationships. In [13, 15], the experiments in Domain A show that sentences matching all or most arguments of a semantic seed for a quaternary relation are the best candidates for mentions of the target relation. The learned quaternary rules exhibit best precision and recall among all rules. The projection rules with three arguments again perform much better than binary rules.

Let $r =< a_1, a_2, ..., a_n >$ be an instance of an n-ary relation R.

Then $T_r$, the tuple-mention set of $r$, is the set of all segments (i.e., sentences) in which mentions of all arguments are detected. $T'_r$ is a subset of $T_r$ containing exactly those segments that actually intensionally refer to the instance of $r$. The ratio $T'_r/T_r$ we call distinctiveness. The more distinctive a seed, the better for precision.

If we have a large number of correct seed instances available, we can use all of them in order to secure a good start of the learning process. If we only have a small set, we need to think about ways to increase the seed fast during the first step without already collecting too many false positives.

Assume we want to build a database of married couples from the web. If we use as seeds some pairs of the form:

<wife: “*Robin Wright*”, husband: “*Sean Penn*”>,

then the seed might also trigger many mentions of other events or relations involving the two persons, i.e., *meeting*, *co-starring*, *fighting* and *separating*.

An effective method for increasing the distinctiveness is the extension of the n-ary relation by increasing the arity of the relation and accordingly of the seed. If wisely selected, the addition of another relevant argument for the first step of the bootstrapping can make the relation more distinctive. In subsequent steps, the original nary relation will be among the selected projections of the initial n+1-ary relation.

Another strategy for increasing the distinctiveness is to use seed examples whose participants are less likely to appear in reports about other relation types besides the target relation. If we select as a seed pair two participants who rarely appear in the data (in our case in the news) we may not get the desired degree of redundancy. Thus we have to restrict this strategy to one participant, e.g., select a seed couple in which only one partner enjoys the desired level of popularity.

We are now going to back up these proposals by measured evidence. To this end we stay with the detection of married couples. We consider the following parameters:

a) arity of the seed example
b) size of the seed set
c) distinctiveness of the seed example

In case of a), two options are taken into account

1. <person1, person2>

2. <person1, person2, marriage year>

In the case b), we assume that including more pairs exhibiting the same relation will help to find the relevant patterns fast. For the case c), we selected couples of which one partner does not appear in the press except as spouse of the more popular partner. We also picked people who had not been married before or after to further increase the distinctiveness of the seed examples. In our experiment we vary the three factors.

In order to avoid data sparseness and unlucky seeds, we choose only prominent persons from the Wikipedia. 313 persons were selected, belonging to the English Wikipedia categories "Times person", "presidents of US", "best drama actor Golden Globe", "best actor of Academy award", "best actress of Academy award" and "best support actor Golden Globe". Our data setup contains 313 documents. We extract 11733 sentences which contain mentionings of two persons. We conducted one experiment for each of the six seed configurations:

**(1)** arity 2, size 1, more distinctive
**(2)** arity 2, size 3, more distinctive
**(3)** arity 3, size 3, more distinctive

To explain our attempt to increase distinctiveness: The seeds contain couples whose spouses married only once, and for which one spouse is not mentioned in other contexts in the Wikipedia, i.e., the US president "Andrew Johnson" and his wife "Eliza McCardle".

**(4)** arity 2, size 3, less distinctive
**(5)** arity 3, size 3, less distinctive

Each person in the above marriage couples have married at least twice and both partners are involved in different mentioned relations, e.g., "Audrey Hepburn" and "Mel Ferrer".

**(6)** arity 3, size 6, mixed distinctiveness

The seed contains three more distinctive and three less distinctive couples with their respective wedding years. Figure 6 illustrates the recall and the precision results with the above six seed configurations. The most striking result is the evidence showing that more distinctive examples achieve much better precision and better recall than less distinctive seeds. Raising the arity for increased distinctiveness results in slight improvements of precision and recall. Increasing the number of seed examples does not help with the Wikipedia data, while the size of the seed set plays a central role for the experiment with Domain B mentioned in Section 2 ([14]) whose corpus exhibits less redundancy and connectivity.

## 4 Negative Seed for Negative Rules

An ideal learning graph contains all instances of the target relation that are mentioned in the data and therefore probably also all inducible rules. As mentioned

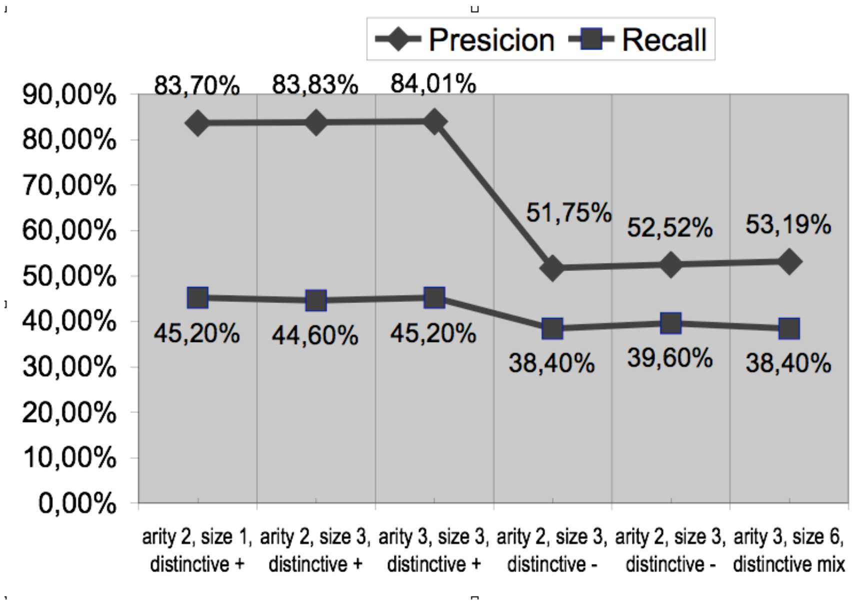



Fig. 6.: arity, size and distinctiveness

in Section 3, the ambiguity potential of the semantic seed is one of the error sources, in particular, those of the simple relations. In the experiment with Domain A ([14]), a systematic evaluation has been conducted for errors. It turned out that modality (17.6%) and rules (14.7%) are two major error resources in addition to factual errors in the content and parsing errors.

Facts or events can be embedded in the scope of modalities that either deny or weaken the truth of the statements. The relevant mentions of the relation may occur within the scope of negation particles, modal verbs, modal adverbials propositional attitude verbs, and other modifications affecting the truth value. In example (9) from Domain corpus A, the facts embedded within the scope of "speculation" are wrong.

**(9)** *The talk has included speculation [that North Korean leader Kim Jong Il and South Korean President Kim Dae-jung might win the Nobel Peace Prize for their step toward reconciliation, the most promising sign of rapprochement since the Korean war ended with a fragile truce in 1953]*

Learned rules can be classified into four categories: good, useless, dangerous and bad. Good rules extract only correct instances. Useless rules are rules that do not contribute any extractions of new instances in the bootstrapping process. Bad rules are those that discover only wrong instances. In Domain A, the bad rules only resulted in 5% of the error instances. Dangerous rules are the rules, which extract both correct and incorrect instances. They exhibit the highest error spreading potential and they caused most of the errors in our experiments. Often they detect semantically related relations whose extensions overlap with the target relation. In the prize award domain, for instance, the patterns headed by the verb "nominate" extract many award winners, because they had also been nominated before being selected. In the marriage domain, some high-ranked dangerous rules are headed by the semantically independent verbs "meet" and "play" because these are two other reported relations in which the spouses often stand. A tempting strategy for preventing bad and dangerous rules is to apply negative rules learned from negative examples as filters of the learned instances. There are two semantically different negative seed sorts

1. dangerous instances:
   seed examples that also represent other relations with the same argument combination as the target relation;
2. wrong facts or events:
   seed examples that are explicitly mentioned as wrong instances in the corpus.

The dangerous instances already help us to learn dangerous rules, while the wrong facts or events will hopefully lead us to learn modalities affecting the truth value. In the following, we present two bootstrapping methods that integrate learning of negative rules. By the first algorithm (Method 1), we learn at first the negative rules from the negative seed and then integrate the negative rules in each positive rule learning iteration, see the following detailed description. It is a conservative method because it learns the negative rules from the initial negative seed set. The number of learned rules grows with the cardinality of the seed set.

**Algorithm 1** Method 1 - Single Step Negative Rule Learning

**Given**: PositiveSeeds, NegativeSeeds

//get negative patterns
**NegativePatterns** = *getPaterns*(NegativeSeeds);

//bootstrapping for positive seeds
**while** (PositiveSeeds not empty) {

  patterns = *getPatterns*(PositiveSeeds);

  //remove the negative patterns
    **patterns = patterns - NegativePatterns ;**

  PositiveSeeds = *getInstance*(patterns);

}

**Algorithm 2** Method 2 - Double Negative Rule Learning

**Given**: PositiveSeeds, NegativeSeeds

//get negative patterns
**NegativePatterns** = *getPaterns*(NegativeSeeds);

//double bootstrapping
**while** (PositiveSeeds not empty) {

PositivePatterns = *getPatterns*(PositiveSeeds);
**PositivePatterns = PositivePatterns  NegativePatterns**;
PositiveSeeds = *getInstance*(PositivePatterns);

//learning for negative patterns
NegativeSeeds = *getInstance*(NegativePatterns);
NegativeSeeds = NegativeSeeds  PositiveSeeds;
**NegativePatterns += getPatterns(NegativeSeeds)**;

}

The second method (Method 2) is an extension of method 1. It allows the bootstrapping of negative seeds. The negative seeds will be the set of instances learned by the negative patterns minus the positive seeds. This method is an eager approach, since the new negative seed can potentially contain positive instances that are not member of the known positive seed. This extended notion of negative seed will cause incorrect negative rules.

We first conduct the experiments in the marriage domain. We consider the following relations as dangerous: "meet", "work-with", "affair", "siblings" and "friendship" and construct a negative seed set of these relations for our learning system. Table 1 shows the experiment results of the six options with the two negative methods in comparison to the previous results shown in Figure 6. The results in Table 1 show that the integration of negative seeds has improved the precision values, in particular for the less distinctive seed examples. Method 2 works a little better than method 1 for precision.

Table 2 presents the number of correct instances extracted before and after integration of the negative seeds. Method 1 slightly reduces the number of correct instances whereas Method 2 really damages recall. This means that the eager approach has also deleted positive seeds during the bootstrapping process.

The second experiment applying Method 1 uses the same corpus for the extraction of prize and award events. The negative seed is an instance of the event "nomination". The results are listed in Table 3. The precision has considerably improved without hurting recall.

For constructing the second kind of negative seed, namely, explicitly negated facts and events mentioned in the corpus, a corpus analysis is needed. We take the corpus for Domain A and try to find the negative examples of Nobel Prize

Table 1.: Precision with Negative Seeds

| methods | without negative | with negative method 1 | with negative method 2 |
|---|---|---|---|
| arity 2, size 1, more distinctive | 83,70% | 96,23% | 97,50% |
| arity 2, size 3, more distinctive | 83,83% | 96,17% | 97,44% |
| arity 3, size 3, more distinctive | 84,01% | 96,71% | 97,44% |
| arity 2, size 3, less distinctive | 51,75% | 96,19% | 96,43% |
| arity 3, size 3, less distinctive | 52,52% | 96,02% | 96,56% |
| arity 3, size 6, mixture | 53,19% | 95,91% | 96,43% |

Table 2.: Correct Results with Negative Seeds

| methods | without negative | with negative method 1 | with negative method 2 |
|---|---|---|---|
| arity 2, size 1, distinctive | 226 | 204 | 117 |
| arity 2, size 3, distinctive | 223 | 201 | 152 |
| arity 3, size 3, distinctive | 226 | 204 | 152 |
| arity 2, size 3, indistinctive | 192 | 202 | 135 |
| arity 3, size 3, indistinctive | 198 | 169 | 138 |
| arity 3, size 6, mixture | 192 | 164 | 135 |

winners. "Rudy Giuliani" is among the frequently mentioned persons who also co-occur with the concept "Nobel Prize".

The mentioning of "Rudy Giuliani" and the "Nobel Prize" is embedded in the scope of the modal adjective "possible":

**(10)** *It's also possible [that O.J. Simpson will find the real killer, that Bill Clinton will enter a monastery and that Rudy Giuliani will win the Nobel Peace Prize.]*

In fact, modalities can be expressed in a variety of ways, e.g., by a noun such as "speculation" in (9), or by modal adjective or adverb such as "possible" like (10) or "never", or even by some ctional contexts provided by films or novels. The linguistic structures embedded in the modality scopes are highlighted by

Table 3.: Negative Seed for prize and award domain

| methods | without negative | with negative |
|---|---|---|
| precision | 69,48% | 74,89% |
| recall | 42,82% | 42,82% |

us with brackets. (10) poses an additional challenge because of irony. Sentence (11) introduces a fictional Nobel Prize winner, "Josiah Bartlett", announced by a TV program. Thus, world knowledge is needed here to resolve the modality.

**(11)** *In <u>NBC</u> "West Wing," [we get President Josiah Bartlett, a Nobel Prize Winner]*

Our current rule-inducing algorithm cannot yet identify and learn the relevant constructions outside the actual mention of the relation instance because this would require a more sophisticated generalization step. Work on extending our rule generalization is in progress.

## 5 Conclusion and Future Work

Curiosity was our original motivation for the systematic analysis of the effects of data and seed properties for the performance of the learning method. After realizing that performance varied drastically among different domains and data sets, we wanted to understand the reasons for this variation. The deeper understanding of the interaction of target relation, seeds and data then lead us to some hypotheses concerning strategies for improving the method. Most of these strategies could already be tested, some of them with positive results.

The implementation and validation of the other proposed strategies require considerable additional efforts, which are now on our plans for future research. Among them are the learning of negative patterns and rules from a small subset of the relation together with its complement. In the case of Prize Awards, Nobel Prize Winners constitute such a closed subset, since all laureates are known. For many other domains such a set can easily be constructed. An example for marriages of movie celebrities is a set of known non-couples among frequently co-occurring pairs of movie stars.

## Acknowledgement

The presented research was partially supported through the project TAKE funded by a grant from the German Federal Ministry of Education and Research (FKZ: 01IW08003) and through the project KomParse funded by the ProFIT programme of the Federal State of Berlin and the EFRE programme of the European Union.

## References

1. S. Abney. Bootstrapping. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 360–367, 2002.
2. S. Abney. Understanding the Yarowsky algorithm. *Computational Linguistics*, 30(3):365–395, 2004.

3. Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries (DL'00)*, San Antonio, TX, June 2000.
4. S. Blohm and P. Cimiano. Using the Web to Reduce Data Sparseness in Pattern-based Information Extraction. *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, September 2007.
5. Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled sata with co-training. In *COLT*, pages 92–100, 1998.
6. Sergey Brin. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, 1998.
7. Mark A. Greenwood and Mark Stevenson. Improving semi-supervised acquisition of relation extraction patterns. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 29–35, Sydney, Australia, July 2006. Association for Computational Linguistics.
8. M.A. Hearst. Automatic Acquisition of Hyponyms om Large Text Corpora. *Proceedings of the Fourteenth International Conference on Computational Linguistics*, 1992.
9. R. Jones. *Learning to Extract Entities from Labeled and Unlabeled Text.* PhD thesis, University of Utah, 2005.
10. Ellen Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049. The AAAI Press/MIT Press, 1996.
11. M. Stevenson and M.A. Greenwood. A Semantic Approach to IE Pattern Induction. *Ann Arbor*, 100, 2005.
12. K. Sudo, S. Sekine, and R. Grishman. An improved extraction pattern representation model for automatic IE pattern acquisition. *Proceedings of ACL 2003*, pages 224–231, 2003.
13. Feiyu Xu. *Bootstrapping Relation Extraction from Semantic Seeds.* Phd-thesis, Saarland University, 2007.
14. Feiyu Xu and Hans Uszkoreit. Minimally supervised learning of relation extraction rules using semantic seeds. A seminar talk at the National Center for Text Mining (NaCTeM), May 2007.
15. Feiyu Xu, Hans Uszkoreit, and Hong Li. Automatic event and relation detection with seeds of varying complexity. In *Proceedings of AAAI 2006 Workshop Event Extraction and Synthesis*, Boston, July 2006.
16. Feiyu Xu, Hans Uszkoreit, and Hong Li. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL07)*, pages 584–591, 2007.
17. Roman Yangarber. *Scenarion Customization for Information Extraction.* Dissertation, Department of Computer Science, Graduate School of Arts and Science, New York University, New York, USA, 2001.
18. D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics Morristown, NJ, USA, 1995.