Learning Entailment Rules for Unary Templates

Idan Szpektor
Department of Computer Science
Bar-Ilan University
Ramat Gan, Israel
`szpekti@macs.biu.ac.il`

Ido Dagan
Department of Computer Science
Bar-Ilan University
Ramat Gan, Israel
`dagan@macs.biu.ac.il`

Abstract

Most work on unsupervised entailment rule acquisition focused on rules between templates with two variables, ignoring *unary rules* - entailment rules between templates with a single variable. In this paper we investigate two approaches for unsupervised learning of such rules and compare the proposed methods with a binary rule learning method. The results show that the learned unary rule-sets outperform the binary rule-set. In addition, a novel directional similarity measure for learning entailment, termed *Balanced-Inclusion*, is the best performing measure.

1 Introduction

In many NLP applications, such as Question Answering (QA) and Information Extraction (IE), it is crucial to recognize whether a specific target meaning is inferred from a text. For example, a QA system has to deduce that "*SCO sued IBM*" is inferred from "*SCO won a lawsuit against IBM*" to answer "*Whom did SCO sue?*". This type of reasoning has been identified as a core semantic inference paradigm by the generic *Textual Entailment* framework (Giampiccolo et al., 2007).

An important type of knowledge needed for such inference is *entailment rules*. An entailment rule specifies a directional inference relation between two *templates*, text patterns with variables, such as 'X win lawsuit against $Y \rightarrow X$ sue Y'. Applying this rule by matching 'X win lawsuit against Y' in the above text allows a QA system to

infer 'X sue Y' and identify "*IBM*", Y's instantiation, as the answer for the above question. Entailment rules capture linguistic and world-knowledge inferences and are used as an important building block within different applications, e.g. (Romano et al., 2006).

One reason for the limited performance of generic semantic inference systems is the lack of broad-scale knowledge-bases of entailment rules (in analog to lexical resources such as WordNet). Supervised learning of broad coverage rule-sets is an arduous task. This sparked intensive research on unsupervised acquisition of entailment rules (and similarly paraphrases) e.g. (Lin and Pantel, 2001; Szpektor et al., 2004; Sekine, 2005).

Most unsupervised entailment rule acquisition methods learn *binary rules*, rules between templates with two variables, ignoring *unary rules*, rules between *unary templates* (templates with only one variable). However, a predicate quite often appears in the text with just a single variable (e.g. intransitive verbs or passives), where inference requires unary rules, e.g. 'X take a nap $\rightarrow X$ sleep' (further motivations in Section 3.1).

In this paper we focus on unsupervised learning of unary entailment rules. Two learning approaches are proposed. In our main approach, rules are learned by measuring how similar the variable instantiations of two templates in a corpus are. In addition to adapting state-of-the-art similarity measures for unary rule learning, we propose a new measure, termed *Balanced-Inclusion*, which balances the notion of directionality in entailment with the common notion of symmetric semantic similarity. In a second approach, unary rules are derived from binary rules learned by state-of-the-art binary rule learning methods.

We tested the various unsupervised unary rule

learning methods, as well as a binary rule learning method, on a test set derived from a standard IE benchmark. This provides the first comparison between the performance of unary and binary rulesets. Several results rise from our evaluation: (a) while most work on unsupervised learning ignored unary rules, all tested unary methods outperformed the binary method; (b) it is better to learn unary rules directly than to derive them from a binary rule-base; (c) our proposed Balanced-Inclusion measure outperformed all other tested methods in terms of F1 measure. Moreover, only Balanced-Inclusion improved F1 score over a baseline inference that does not use entailment rules at all .

2 Background

This section reviews relevant distributional similarity measures, both symmetric and directional, which were applied for either lexical similarity or unsupervised entailment rule learning.

Distributional similarity measures follow the Distributional Hypothesis, which states that words that occur in the same contexts tend to have similar meanings (Harris, 1954). Various measures were proposed in the literature for assessing such similarity between two words, u and v. Given a word q, its set of features F_q and feature weights $w_q(f)$ for $f \in F_q$, a common symmetric similarity measure is Lin similarity (Lin, 1998a):

$$Lin(u,v) = \frac{\sum_{f \in F_u \cap F_v}[w_u(f) + w_v(f)]}{\sum_{f \in F_u} w_u(f) + \sum_{f \in F_v} w_v(f)}$$

where the weight of each feature is the pointwise mutual information (pmi) between the word and the feature: $w_q(f) = log[\frac{Pr(f|q)}{Pr(f)}]$.

Weeds and Weir (2003) proposed to measure the symmetric similarity between two words by averaging two directional (asymmetric) scores: the coverage of each word's features by the other. The coverage of u by v is measured by:

$$Cover(u,v) = \frac{\sum_{f \in F_u \cap F_v} w_u(f)}{\sum_{f \in F_u} w_u(f)}$$

The average can be arithmetic or harmonic:
$$WeedsA(u,v) = \frac{1}{2}[Cover(u,v) + Cover(v,u)]$$

$$WeedsH(u,v) = \frac{2 \cdot Cover(u,v) \cdot Cover(v,u)}{Cover(u,v) + Cover(v,u)}$$

Weeds et al. also used pmi for feature weights.

Binary rule learning algorithms adopted such lexical similarity approaches for learning rules between templates, where the features of each template are its variable instantiations in a corpus, such as $\{X=\text{'}SCO\text{'}, Y=\text{'}IBM\text{'}\}$ for the example in Section 1. Some works focused on learning rules from *comparable corpora*, containing comparable documents such as different news articles from the same date on the same topic (Barzilay and Lee, 2003; Ibrahim et al., 2003). Such corpora are highly informative for identifying variations of the same meaning, since, typically, when variable instantiations are shared across comparable documents the same predicates are described. However, it is hard to collect broad-scale comparable corpora, as the majority of texts are non-comparable.

A complementary approach is learning from the abundant regular, non-comparable, corpora. Yet, in such corpora it is harder to recognize variations of the same predicate. The DIRT algorithm (Lin and Pantel, 2001) learns non-directional binary rules for templates that are paths in a dependency parse-tree between two noun variables X and Y. The similarity between two templates t and t' is the geometric average:

$$DIRT(t,t') = \sqrt{Lin_x(t,t') \cdot Lin_y(t,t')}$$

where Lin_x is the Lin similarity between X's instantiations of t and X's instantiations of t' in a corpus (equivalently for Lin_y). Some works take the combination of the two variable instantiations in each template occurrence as a single complex feature, e.g. $\{X\text{-}Y=\text{'}SCO\text{-}IBM\text{'}\}$, and compare between these complex features of t and t' (Ravichandran and Hovy, 2002; Szpektor et al., 2004; Sekine, 2005).

Directional Measures Most rule learning methods apply a symmetric similarity measure between two templates, viewing them as paraphrasing each other. However, entailment is in general a directional relation. For example, 'X acquire Y → X own Y' and 'countersuit against X → lawsuit against X'.

(Weeds and Weir, 2003) propose a directional measure for learning hyponymy between two words, 'l → r', by giving more weight to the coverage of the features of l by r (with $\alpha > \frac{1}{2}$):

$$WeedsD(l,r) = \alpha Cover(l,r) + (1-\alpha)Cover(r,l)$$

When $\alpha=1$, this measure degenerates into $Cover(l,r)$, termed $Precision(l,r)$. With

$Precision(l, r)$ we obtain a "soft" version of the inclusion hypothesis presented in (Geffet and Dagan, 2005), which expects l to entail r if the "important" features of l appear also in r.

Similarly, the LEDIR algorithm (Bhagat et al., 2007) identifies the entailment direction between two binary templates, l and r, which participate in a relation learned by (the symmetric) DIRT, by measuring the proportion of instantiations of l that are covered by the instantiations of r.

As far as we know, only (Shinyama et al., 2002) and (Pekar, 2006) learn rules between unary templates. However, (Shinyama et al., 2002) relies on comparable corpora for identifying paraphrases and simply takes any two templates from comparable sentences that share a named entity instantiation to be paraphrases. Such approach is not feasible for non-comparable corpora where statistical measurement is required. (Pekar, 2006) learns rules only between templates related by local discourse (information from different documents is ignored). In addition, their template structure is limited to only verbs and their direct syntactic arguments, which may yield incorrect rules, e.g. for light verbs (see Section 5.2). To overcome this limitation, we use a more expressive template structure.

3 Learning Unary Entailment Rules

3.1 Motivations

Most unsupervised rule learning algorithms focused on learning binary entailment rules. However, using binary rules for inference is not enough. First, a predicate that can have multiple arguments may still occur with only one of its arguments. For example, in "*The acquisition of TCA was successful*", '*TCA*' is the only argument of '*acquisition*'. Second, some predicate expressions are unary by nature. For example, modifiers, such as 'the elected X', or intransitive verbs. In addition, it appears more tractable to learn all variations for each argument of a predicate separately than to learn them for combinations of argument pairs.

For these reasons, it seems that unary rule learning should be addressed in addition to binary rule learning. We are further motivated by the fact that some (mostly supervised) works in IE found learning unary templates useful for recognizing relevant named entities (Riloff, 1996; Sudo et al., 2003; Shinyama and Sekine, 2006), though they did not attempt to learn generic knowledge bases of entailment rules.

This paper investigates acquisition of unary entailment rules from regular non-comparable corpora. We first describe the structure of unary templates and then explore two conceivable approaches for learning unary rules. The first approach directly assesses the relation between two given templates based on the similarity of their instantiations in the corpus. The second approach, which was also mentioned in (Iftene and Balahur-Dobrescu, 2007), derives unary rules from learned binary rules.

3.2 Unary Template Structure

To learn unary rules we first need to define their structure. In this paper we work at the syntactic representation level. Texts are represented by dependency parse trees (using the Minipar parser (Lin, 1998b)) and templates by parse sub-trees.

Given a dependency parse tree, any sub-tree can be a candidate template, setting some of its nodes as variables (Sudo et al., 2003). However, the number of possible templates is exponential in the size of the sentence. In the binary rule learning literature, the main solution for exhaustively learning all rules between any pair of templates in a given corpus is to restrict the structure of templates. Typically, a template is restricted to be a path in a parse tree between two variable nodes (Lin and Pantel, 2001; Ibrahim et al., 2003).

Following this approach, we chose the structure of unary templates to be paths as well, where one end of the path is the template's variable. However, paths with one variable have more expressive power than paths between two variables, since the combination of two unary paths may generate a binary template that is not a path. For example, the combination of 'X call indictable' and 'call Y indictable' is the template 'X call Y indictable', which is not a path between X and Y.

For every noun node v in a parsed sentence, we generate templates with v as a variable as follows:

1. Traverse the path from v towards the root of the parse tree. Whenever a candidate predicate is encountered (any noun, adjective or verb) the path from that node to v is taken as a template. We stop when the first verb or clause boundary (e.g. a relative clause) is encountered, which typically represent the syntactic boundary of a specific predicate.

2. To enable templates with control verbs and light verbs, e.g. 'X help preventing', 'X make noise', whenever a verb is encountered we generate templates that are paths between v and the verb's modifiers, either objects, prepositional complements or infinite or gerund verb forms (paths ending at stop words, e.g. pronouns, are not generated).

3. To capture noun modifiers that act as predicates, e.g. 'the losing X', we extract template paths between v and each of its modifiers, nouns or adjectives, that are derived from a verb. We use the Catvar database to identify verb derivations (Habash and Dorr, 2003).

As an example for the procedure, the templates extracted from the sentence *"The losing party played it safe"* with '*party*' as the variable are: 'losing X', 'X play' and 'X play safe'.

3.3 Direct Learning of Unary Rules

We applied the lexical similarity measures presented in Section 2 for unary rule learning. Each argument instantiation of template t in the corpus is taken as a feature f, and the pmi between t and f is used for the feature's weight. We first adapted DIRT for unary templates (*unary-DIRT*, applying Lin-similarity to the single feature vector), as well as its output filtering by LEDIR. The various Weeds measures were also applied[1]: symmetric arithmetic average, symmetric harmonic average, weighted arithmetic average and Precision.

After initial analysis, we found that given a right hand side template r, symmetric measures such as Lin (in DIRT) generally tend to prefer (score higher) relations $\langle l, r \rangle$ in which l and r are related but do not necessarily participate in an entailment or equivalence relation, e.g. the wrong rule 'kill X $\leftrightarrow$ injure X'.

On the other hand, directional measures such as Weeds Precision tend to prefer directional rules in which the entailing template is infrequent. If an infrequent template has common instantiations with another template, the coverage of its features is typically high, whether or not an entailment relation exists between the two templates. This behavior generates high-score incorrect rules.

Based on this analysis, we propose a new measure that balances the two behaviors, termed

---

Balanced-Inclusion (**BInc**). BInc identifies entailing templates based on a directional measure but penalizes infrequent templates using a symmetric measure:

$$BInc(l, r) = \sqrt{Lin(l, r) \cdot Precision(l, r)}$$

3.4 Deriving Unary Rules From Binary Rules

An alternative way to learn unary rules is to first learn binary entailment rules and then derive unary rules from them. We derive unary rules from a given binary rule-base in two steps. First, for each binary rule, we generate all possible unary rules that are part of that rule (each unary template is extracted following the same procedure described in Section 3.2). For example, from 'X find solution to $Y \rightarrow X$ solve Y' we generate the unary rules 'X find $\rightarrow X$ solve', 'X find solution $\rightarrow X$ solve', 'solution to $Y \rightarrow$ solve Y' and 'find solution to $Y \rightarrow$ solve Y'. The score of each generated rule is set to be the score of the original binary rule.

The same unary rule can be derived from different binary rules. For example, 'hire $Y \rightarrow$ employ Y' is derived both from 'X hire $Y \rightarrow X$ employ Y' and 'hire Y for $Z \rightarrow$ employ Y for Z', having a different score from each original binary rule. The second step of the algorithm aggregates the different scores yielded for each derived rule to produce the final rule score. Three aggregation functions were tested: sum (*Derived-Sum*), average (*Derived-Avg*) and maximum (*Derived-Max*).

4 Experimental Setup

We want to evaluate learned unary and binary rule bases by their utility for NLP applications through assessing the validity of inferences that are performed in practice using the rule base.

To perform such experiments, we need a test-set of *seed templates*, which correspond to a set of target predicates, and a corpus annotated with all argument mentions of each predicate. The evaluation assesses the correctness of all argument extractions, which are obtained by matching in the corpus either the seed templates or templates that entail them according to the rule-base (the latter corresponds to *rule-application*).

Following (Szpektor et al., 2008), we found the ACE 2005 event training set[2] useful for this purpose. This standard IE dataset includes 33 types of event predicates such as *Injure, Sue* and *Divorce*.

All event mentions are annotated in the corpus, including the instantiated arguments of the predicate. ACE guidelines specify for each event its possible arguments, each associated with a semantic role. For instance, some of the *Injure* event arguments are *Agent*, *Victim* and *Time*.

To utilize the ACE dataset for evaluating entailment rule applications, we manually represented each ACE event predicate by unary seed templates. For example, the seed templates for *Injure* are 'A injure', 'injure V' and 'injure in T'. We mapped each event role annotation to the corresponding seed template variable, e.g. '*Agent*' to A and '*Victim*' to V in the above example. Templates are matched using a syntactic matcher that handles simple morpho-syntactic phenomena, as in (Szpektor and Dagan, 2007). A rule application is considered correct if the matched argument is annotated by the corresponding ACE role.

For testing binary rule-bases, we automatically generated binary seed templates from any two unary seeds that share the same predicate. For example, for *Injure* the binary seeds 'A injure V', 'A injure in T' and 'injure V in T' were automatically generated from the above unary seeds.

We performed two adaptations to the ACE dataset to fit it better to our evaluation needs. First, our evaluation aims at assessing the correctness of inferring a specific target semantic meaning, which is denoted by a specific predicate, using rules. Thus, four events that correspond ambiguously to multiple distinct predicates were ignored. For instance, the *Transfer-Money* event refers to both *donating* and *lending* money, and thus annotations of this event cannot be mapped to a specific seed template. We also omitted 3 events with less than 10 mentions, and were left with 26 events (6380 argument mentions).

Additionally, we regard all entailing mentions under the textual entailment definition as correct. However, event mentions are annotated as correct in ACE only if they explicitly describe the target event. For instance, a Divorce mention does entail a preceding marriage event but it does not explicitly describe it, and thus it is not annotated as a *Marry* event. To better utilize the ACE dataset, we considered for a target event the annotations of other events that entail it as being correct as well. We note that each argument was considered separately. For example, we marked a mention of a divorced person as entailing the marriage of that person, but did not consider the place and time of the divorce act to be those of the marriage .

5   Results and Analysis

We implemented the unary rule learning algorithms described in Section 3 and the binary DIRT algorithm (Lin and Pantel, 2001). We executed each method over the Reuters RCV1 corpus[3], learning for each template r in the corpus the top 100 rules in which r is entailed by another template l, '$l \rightarrow r$'. All rules were learned in canonical form (Szpektor and Dagan, 2007). The rule-base learned by binary DIRT was taken as the input for deriving unary rules from binary rules.

The performance of each acquired rule-base was measured for each ACE event. We measured the percentage of correct argument mentions extracted out of all correct argument mentions annotated for the event (recall) and out of all argument mentions extracted for the event (precision). We also measured F1, their harmonic average, and report macro average *Recall*, *Precision* and *F1* over the 26 event types.

No threshold setting mechanism is suggested in the literature for the scores of the different algorithms, especially since rules for different right hand side templates have different score ranges. Thus, we follow common evaluation practice (Lin and Pantel, 2001; Geffet and Dagan, 2005) and test each learned rule-set by taking the top K rules for each seed template, where K ranges from 0 to 100. When K=0, no rules are used and mentions are extracted only by direct matching of seed templates.

Our rule application setting provides a rather simplistic IE system (for example, no named entity recognition or approximate template matching). It is thus useful for comparing different rule-bases, though the absolute extraction figures do not reflect the full potential of the rules. In Secion 5.2 we analyze the full-system's errors to isolate the rules' contribution to overall system performance.

5.1   Results

In this section we focus on the best performing variations of each algorithm type: binary DIRT, unary DIRT, unary Weeds Harmonic, BInc and *Derived-Avg*. We omitted the results of methods that were clearly inferior to others: (a) $WeedsA$, $WeedsD$ and $Weeds\text{-}Precision$ did not increase Recall over not using rules because rules with very

---

infrequent templates scored highest and arithmetic averaging could not balance well these high scores; (b) out of the methods for deriving unary rules from binary rule-bases, *Derived-Avg* performed best; (c) filtering with (the directional) LEDIR did not improve the performance of unary DIRT.

Figure 1 presents Recall, Precision and F1 of the methods for different cutoff points. First, we observe that even when matching only the seed templates (K=0), unary seeds outperform the binary seeds in terms of both Precision and Recall. This surprising behavior is consistent through all rule cutoff points: all unary learning algorithms perform better than binary DIRT in all parameters. The inferior behavior of binary DIRT is analyzed in Section 5.2.

The graphs show that symmetric unary approaches substantially increase recall, but dramatically decrease precision already at the top 10 rules. As a result, F1 only decreases for these methods. Lin similarity (DIRT) and Weeds-Harmonic show similar behaviors. They consistently outperform *Derived-Avg*. One reason for this is that incorrect unary rules may be derived even from correct binary rules. For example, from 'X gain seat on $Y \rightarrow$ elect X to Y' the incorrect unary rule 'X gain $\rightarrow$ elect X' is also generated. This problem is less frequent when unary rules are directly scored based on their corpus statistics.

The directional measure of BInc yields a more accurate rule-base, as can be seen by the much slower precision reduction rate compared to the other algorithms. As a result, it is the only algorithm that improves over the F1 baseline of K=0, with the best cutoff point at K=20. BInc's recall increases moderately compared to other unary learning approaches, but it is still substantially better than not using rules (a relative recall increase of 50% already at K=10). We found that many of the correct mentions missed by BInc but identified by other methods are due to occasional extractions of incorrect frequent rules, such as partial templates (see Section 5.2). This is reflected in the very low precision of the other methods. On the other hand, some correct rules were only learned by BInc, e.g. 'countersuit against $X \rightarrow X$ sue' and 'X take wife $\rightarrow X$ marry'.

When only one argument is annotated for a specific event mention (28% of ACE predicate mentions, which account for 15% of all annotated arguments), binary rules either miss that mention, or
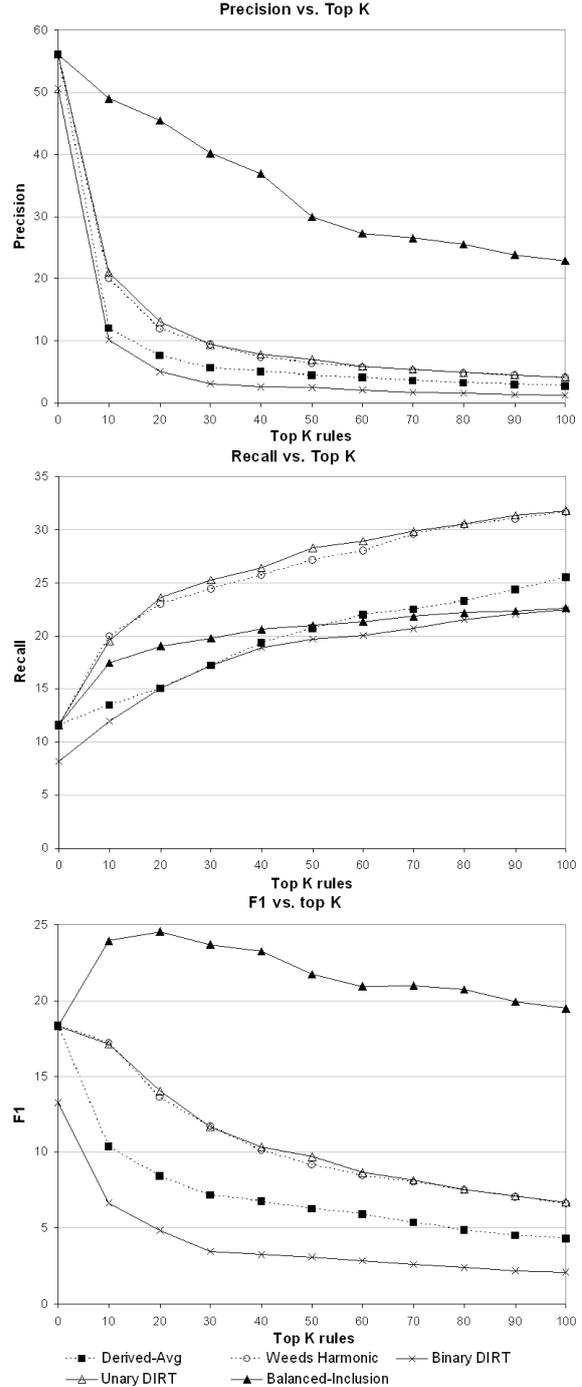

Figure 1: Average Precision, Recall and F1 at different top K rule cutoff points.

extract both the correct argument and another incorrect one. To neutralize this bias, we also tested the various methods only on event mentions annotated with two or more arguments and obtained similar results to those presented for all mentions. This further emphasizes the general advantage of using unary rules over binary rules.

5.2 Analysis

Binary-DIRT We analyzed incorrect rules both for binary-DIRT and BInc by randomly sampling, for each algorithm, 200 rules that extracted incorrect mentions. We manually classified each rule '$l \rightarrow r$' as either: (a) *Correct* - the rule is valid in some contexts of the event but extracted some incorrect mentions; (b) *Partial Template* - l is only a part of a correct template that entails r. For example, learning 'X decide $\rightarrow X$ meet' instead of 'X decide to meet $\rightarrow X$ meet'; (e) *Incorrect* - other incorrect rules, e.g. 'charge $X \rightarrow$ convict X'.

Table 1 summarizes the analysis and demonstrates two problems of binary-DIRT. First, relative to BInc, it tends to learn incorrect rules for high frequency templates, and therefore extracted many more incorrect mentions for the same number of incorrect rules. Second, a large percentage of incorrect mentions extracted are due to partial templates at the rule left-hand-side. Such rules are leaned because many binary templates have a more complex structure than paths between arguments. As explained in Section 3.2 the unary template structure we use is more expressive, enabling to learn the correct rules. For example, BInc learned 'take Y into custody $\rightarrow$ arrest Y' while binary-DIRT learned 'X take $Y \rightarrow X$ arrest Y'.

System Level Analysis We manually analyzed the reasons for false positives (incorrect extractions) and false negatives (missed extractions) of BInc, at its best performing cutoff point (K=20), by sampling 200 extractions of each type.

From the false positives analysis (Table 2) we see that 39% of the errors are due to incorrect rules. The main reasons for learning such rules are those discussed in Section 3.3: (a) related templates that are not entailing; (b) infrequent templates. All learning methods suffer from these issues. As was shown by our results, BInc provides a first step towards reducing these problems. Yet, these issues require further research.

Apart from incorrectly learned rules, incorrect template matching (e.g. due to parse errors) and context mismatch contribute together 46% of the errors. Context mismatches occur when the entailing template is matched in inappropriate contexts. For example, 'slam $X \rightarrow$ attack X' should not be applied when X is a ball, only when it is a person. The rule-set net effect on system precision is better estimated by removing these errors and fixing the annotation errors, which yields 72% precision.

	Binary DIRT		Balanced Inclusion	
Correct	16	(70)	38	(91)
Partial Template	27	(2665)	6	(81)
Incorrect	157	(2584)	156	(787)
Total	200	(5319)	200	(959)

Table 1: Rule type distribution of a sample of 200 rules that extracted *incorrect* mentions. The corresponding numbers of incorrect mentions extracted by the sampled rules is shown in parentheses.

Reason	% mentions
Incorrect Rule learned	39.0
Context mismatch	27.0
Match error	19.0
Annotation problem	15.0

Table 2: Distribution of reasons for false positives (incorrect argument extractions) by BInc at K=20.

Reason	% mentions
Rule not learned	61.5
Match error	25.0
Discourse analysis needed	12.0
Argument is predicative	1.5

Table 3: Distribution of reasons for false negatives (missed argument mentions) by BInc at K=20.

Table 3 presents the analysis of false negatives. First, we note that 12% of the arguments cannot be extracted by rules alone, due to necessary discourse analysis. Thus, a recall upper bound for entailment rules is 88%. Many missed extractions are due to rules that were not learned (61.5%). However, 25% of the mentions were missed because of incorrect syntactic matching of correctly learned rules. By assuming correct matches in these cases we isolate the recall of the rule-set (along with the seeds), which yields 39% recall.

6 Conclusions

We presented two approaches for unsupervised acquisition of unary entailment rules from regular (non-comparable) corpora. In the first approach, rules are directly learned based on distributional similarity measures. The second approach derives unary rules from a given rule-base of binary rules. Under the first approach we proposed a novel directional measure for scoring entailment rules, termed Balanced-Inclusion.

We tested the different approaches utilizing a standard IE test-set and compared them to binary rule learning. Our results suggest the advantage of learning unary rules: (a) unary rule-bases perform

better than binary rules; (b) it is better to directly learn unary rules than to derive them from binary rule-bases. In addition, the Balanced-Inclusion measure outperformed all other tested methods.

In future work, we plan to explore additional unary template structures and similarity scores, and to improve rule application utilizing context matching methods such as (Szpektor et al., 2008).

Acknowledgements

References

Barzilay, Regina and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of HLT-NAACL*.

Bhagat, Rahul, Patrick Pantel, and Eduard Hovy. 2007. Ledir: An unsupervised algorithm for learning directionality of inference rules. In *Proceedings of EMNLP*.

Geffet, Maayan and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of ACL*.

Giampiccolo, Danilo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of WTEP*.

Habash, Nizar and Bonnie Dorr. 2003. A categorial variation database for english. In *Proceedings of NACL*.

Harris, Z. 1954. Distributional structure. *Word*, 10(23):146–162.

Ibrahim, Ali, Boris Katz, and Jimmy Lin. 2003. Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of IWP*.

Iftene, Adrian and Alexandra Balahur-Dobrescu. 2007. Hypothesis transformation and semantic variability rules used in recognizing textual entailment. In *Proceedings of WTEP*.

Lin, Dekang and Patrick Pantel. 2001. Discovery of inference rules for question answering. In *Natural Language Engineering*, volume 7(4), pages 343–360.

Lin, Dekang. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL*.

Lin, Dekang. 1998b. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on Evaluation of Parsing Systems at LREC*.

Pekar, Viktor. 2006. Acquisition of verb entailment from text. In *Proceedings of NAACL*.

Ravichandran, Deepak and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL*.

Riloff, Ellen. 1996. Automatically generating extraction patterns from untagged text. In *AAAI/IAAI, Vol. 2*, pages 1044–1049.

Romano, Lorenza, Milen Kouylekov, Idan Szpektor, Ido Dagan, and Alberto Lavelli. 2006. Investigating a generic paraphrase-based approach for relation extraction. In *Proceedings of EACL*.

Sekine, Satoshi. 2005. Automatic paraphrase discovery based on context and keywords between ne pairs. In *Proceedings of IWP*.

Shinyama, Yusuke and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of HLT-NAACL*.

Shinyama, Yusuke, Satoshi Sekine, Sudo Kiyoshi, and Ralph Grishman. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of HLT*.

Sudo, Kiyoshi, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of ACL*.

Szpektor, Idan and Ido Dagan. 2007. Learning canonical forms of entailment rules. In *Proceedings of RANLP*.

Szpektor, Idan, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP*.

Szpektor, Idan, Ido Dagan, Roy Bar Haim, and Jacob Goldberger. 2008. Contextual preferences. In *Proceedings of ACL*.

Weeds, Julie and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of EMNLP*.