A Web-based Collaborative Evaluation Tool for Automatically Learned Relation Extraction Patterns

Leonhard Hennig, Hong Li, Sebastian Krause, Feiyu Xu, and Hans Uszkoreit
Language Technology Lab, DFKI
Berlin, Germany
{leonhard.hennig,lihong,skrause,feiyu,uszkoreit}@dfki.de

Abstract

Patterns extracted from dependency parses of sentences are a major source of knowledge for most state-of-the-art relation extraction systems, but can be of low quality in distantly supervised settings. We present a linguistic annotation tool that allows human experts to analyze and categorize automatically learned patterns, and to identify common error classes. The annotations can be used to create datasets that enable machine learning approaches to pattern quality estimation. We also present an experimental pattern error analysis for three semantic relations, where we find that between 24% and 61% of the learned dependency patterns are defective due to preprocessing or parsing errors, or due to violations of the distant supervision assumption.

1 Introduction

Dependency parse trees of sentences have been shown to be very useful structures for relation extraction (RE), since they often capture syntactic and semantic properties of a relation and its arguments more compactly than more surface-oriented representations (Grishman, 2012). Typically, shortest-path or similar algorithms are used to extract a pattern from a sentence's dependency parse that connects the relation's arguments. Such patterns can be directly applied to parsed texts to identify novel instances of a relation (Krause et al., 2012), or they can be used as features in a supervised learning approach (Mintz et al., 2009). They are also useful by themselves, as linguistic resources that capture the different ways in which a given human language expresses semantic relations (Uszkoreit and Xu, 2013).

In recent years, distant supervision has become a very important approach to relation extraction (Mintz et al., 2009; Surdeanu et al., 2012; Ritter et al., 2013), due to the availability of large-scale structured knowledge bases such as Freebase (Bollacker et al., 2008). While typically yielding a high recall of relation mentions, distant supervision makes several strong assumptions that may significantly affect the quality of extracted dependency patterns. First, it assumes that for each relation tuple $r_i(e_{i_1}, \ldots, e_{i_k})$ in a knowledge base, every sentence containing mentions of $e_{i_1}, \ldots, e_{i_k}$ (or a subset thereof) expresses the relation r_i (Surdeanu et al., 2012). This assumption typically does not hold for most sentences, i.e., entity mentions may co-occur without the sentence expressing the target relation. Dependency patterns extracted from such sentences should be discarded to improve the precision of an RE system. Furthermore, distant supervision assumes that the knowledge base is complete: entity mention co-occurrences with no known relations are ignored or treated as negative training examples, lowering the discriminative capabilities of a learned model (Ritter et al., 2013).

Automatically estimating the quality of extracted patterns, e.g., by using data-driven statistical metrics, or by learning weights in a supervised setting, leads to indirect measures of pattern quality, but tells us only very little about the (grammatical) correctness and the semantic appropriateness of the patterns themselves. We are hence interested in a more direct, expert-driven analysis of dependency patterns and their properties, which will hopefully guide us towards better automatic quality metrics. To this end, we have developed a linguistic annotation tool, *PatternJudge*, that allows human experts to evaluate relation-specific dependency patterns and their associated source sentences. Our contributions in this paper are:

- We present a linguistic annotation tool for human expert-driven quality control of dependency patterns (Section 3)

Proceedings of ACL-IJCNLP 2015 System Demonstrations, pages 43–48,
Beijing, China, July 26-31, 2015.

- We describe an annotation process for pattern evaluation and the guidelines we developed for it (Section 4)
- We present and discuss common error classes observed in an initial study of three semantic relations (Section 5)

2 Pattern Extraction

In this section, we briefly describe our approach for extracting relation-specific dependency patterns in a distantly supervised setting, called Web-DARE (Krause et al., 2012). In contrast to most other approaches, we consider not only binary, but arbitrary n-ary relations, with $n >= 2$. For example, we can define a 4-ary *marriage* relation with the spouses as essential (required) arguments, and optional arguments such as the wedding date and location. Given a knowledge base (KB) containing such relations and their arguments, we select a set of seed relation instances from the KB. We then collect sentences from a large text corpus that mention at least the essential arguments of a given seed relation instance.

Sentences are preprocessed with a standard NLP pipeline, including tokenization, named entity recognition (NER) and linking, lemmatization, part-of-speech tagging and word sense disambiguation (WSD).[1] We also apply a dependency parser producing Stanford dependency relations. Given a preprocessed sentence and the seed relation instance which matches this sentence, the pattern extraction algorithm first identifies the argument mentions of the seed relation instance occurring in the sentence, and then determines and composes the set of shortest paths connecting the arguments in the dependency parse in a bottom-up manner. Figure 1 visualizes the pattern extraction process for an example sentence expressing the *marriage* relation. The extracted pattern is shown in attribute-value-matrix (AVM) notation in Figure 1c. For more details on the algorithm we refer the interested reader to the DARE pattern extraction method described in Xu et al. (2007).

3 Evaluation tool – PatternJudge

To facilitate the manual evaluation of dependency patterns, we have developed a web-based annotation tool, dubbed *PatternJudge*. With *PatternJudge*, annotators can inspect patterns and source sentences for a given relation, and evaluate their grammatical and semantic correctness. The tool is realized as a browser-based client with a back end web server for data management. It is available online at *http://sargraph.dfki.de/pattern_judge*.

Figure 2 shows a screen shot of the user interface. The interface is split into three main components. The left part displays a list of available relations and patterns, and allows searching for specific patterns or sentences. The center part visualizes the currently selected dependency pattern in AVM notation. In this notation, the INPUT element contains the dependency pattern, and the OUTPUT element lists the relation arguments extracted by this pattern. In the example pattern shown in Figure 2, these correspond to the spouses and the wedding date. Thus, the patterns also contain the semantic role label information of the target relation for the corresponding linguistic arguments, which is not included in most traditional pattern extraction approaches (e.g., Stevenson and Greenwood (2005)).

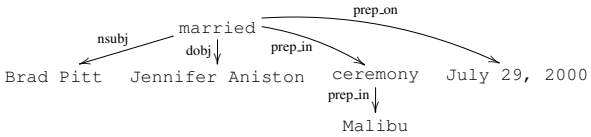
The area below the representation of the pattern lists the source sentences that it was observed in, as well as some statistics about the frequency of the pattern. Sentences are formatted to highlight the important elements of the pattern. Relation arguments are marked in red, content words occurring in the pattern are marked in blue. Listing the source sentences is important because it enables the human expert to verify both the extracted dependency pattern (e.g., to detect a parse error), and the semantic correctness of the pattern, i.e., whether the sentences express the target relation.

The annotation tab on the right-hand side collects the human expert's feedback on the quality of the selected pattern. Currently available options include labeling the pattern as "CORRECT", "CORRECT, BUT TOO SPECIFIC", "INCORRECT" or "UNCERTAIN/DON'T KNOW". We describe the intended scope and meaning of these feedback categories in Section 4. Note that this set of categories is not fixed, but simply reflects what we have found to be useful distinctions thus far for annotating patterns. Annotators can also provide a comment, and, if desired, view the annotations and comments of previous annotators of this pattern. Since multiple experts can collaboratively annotate the same pattern, these comments are

[1] We use the Stanford CoreNLP pipeline (`nlp.stanford.edu/software/corenlp.shtml`), and our own implementation of Babelfy (`babelfy.org`) for WSD and entity linking.

Brad Pitt married Jennifer Aniston in a private wedding ceremony in Malibu on July 29, 2000.

(a) Sentence with a mention of the *marriage* relation.



(b) Dependency pattern extracted from the sentence in (a).

$$\begin{bmatrix} \text{head} & \begin{bmatrix} \text{lemma} & \texttt{marry} \\ \text{POS} & \text{V} \end{bmatrix} \\ \text{dobj} & \begin{bmatrix} \text{type} & \text{person} \\ \text{role} & \text{SPOUSE2} \end{bmatrix} \\ \text{prep\_in} & \begin{bmatrix} \text{head} & \begin{bmatrix} \text{lemma} & \texttt{ceremony} \\ \text{POS} & \text{N} \end{bmatrix} \\ \text{prep\_in} & \begin{bmatrix} \text{type} & \text{location} \\ \text{role} & \text{CEREMONY} \end{bmatrix} \end{bmatrix} \\ \text{prep\_on} & \begin{bmatrix} \text{type} & \text{date} \\ \text{role} & \text{FROM} \end{bmatrix} \\ \text{nsubj} & \begin{bmatrix} \text{type} & \text{person} \\ \text{role} & \text{SPOUSE1} \end{bmatrix} \end{bmatrix}$$

(c) Generalized dependency pattern derived from (b).

Figure 1: Data flow for gathering dependency patterns from distantly labeled text.

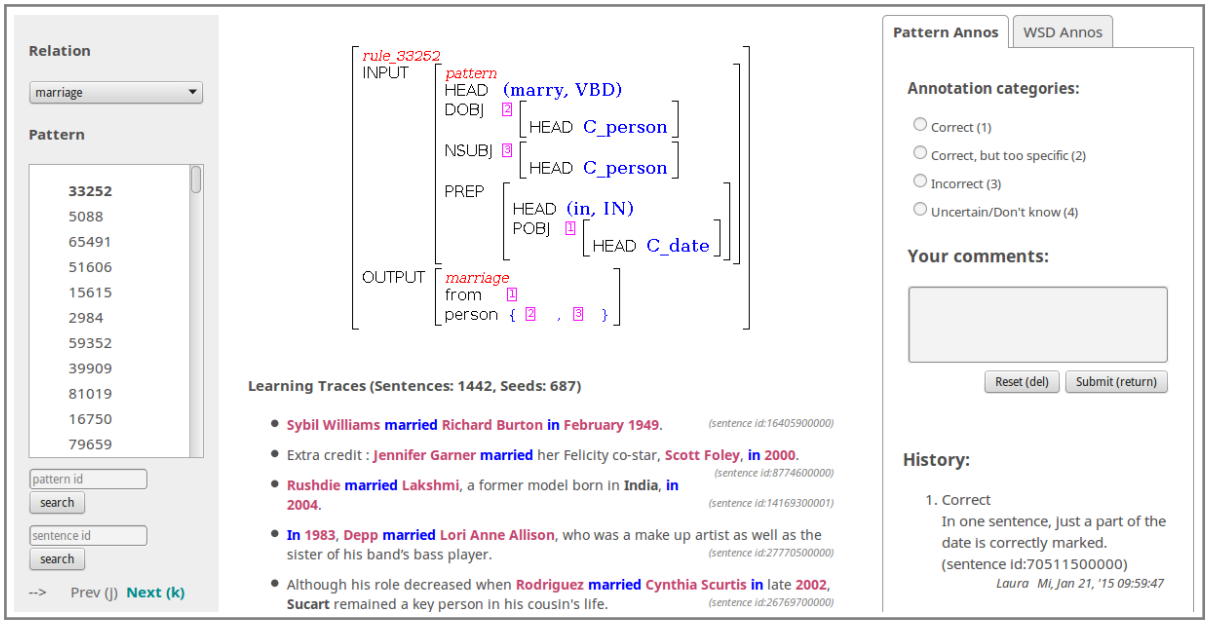


Figure 2: User interface of the *PatternJudge* tool. The tool allows annotators to judge the quality of automatically learned dependency patterns.

mainly used for discussion and clarification, but also for adding error class information in cases where an annotator decided to label a pattern as "INCORRECT".

In a separate tab (not shown in the Figure), annotators can inspect the word senses of the pattern's lemmas. Per lemma, we display a distribution over word senses, since the sentence-level WSD decisions may differ from each other. Annotators can use this view to label the correct word senses for a pattern. Word senses are directly linked to BabelNet[2] for reference. The *PatternJudge* tool also includes a basic user management component to keep track of different annotators, and for undoing or updating previous judgments. All pattern judgments are persisted in a NoSQL data base, and can be exported to CSV or other standard formats for statistical analysis.

4 Expert-driven quality control

We use the *PatternJudge* tool for an experimental analysis of dependency patterns. The analysis has two major goals: to validate interesting, productive dependency patterns, and to identify common error classes of defective patterns. In this section,

[2] http://babelnet.org/

we describe the guidelines that we developed for the manual evaluation process, and the experimental dataset. We report the results of our analysis in Section 5.

4.1 Quality control guidelines

We define three qualitative categories, "CORRECT", "CORRECT, BUT TOO SPECIFIC" and "INCORRECT", as well as a set of annotation guidelines for the evaluation of dependency patterns. We label a relation-specific pattern as "CORRECT" if it is grammatically and semantically correct. A pattern is grammatically correct if there are no parsing or other preprocessing errors, and it is semantically correct if its source sentences express the target relation. Correspondingly, we label a dependency pattern as "INCORRECT" if it is grammatically incorrect, or if its sentences do not express the target relation. Typically, the annotators aim to identify one or more of the error classes discussed in Section 5 to decide whether a pattern is incorrect.

For deciding whether a sentence expresses a given relation, we use the ACE annotation guidelines' conceptual definition of relations and their mentions (Doddington et al., 2004), and define the semantics of relations based on Freebase descriptions. In contrast to the ACE tasks, we also consider n-ary relations in addition to binary relations. Sentences must express the target relation explicitly, e.g., *"Obama was awarded the Nobel Peace Prize."* explicitly expresses the relation *award honor*. We treat implicit mentions as semantically incorrect, e.g., the previous example only implies an *award nomination*.

A third feedback category, "CORRECT, BUT TOO SPECIFIC", was added based on our initial analysis of the dataset, and applies to dependency patterns mostly found in the long tail of the frequency distribution. Too specific patterns, while both grammatically and semantically correct, are patterns that are overly complex and / or include irrelevant parts of the sentence specific to a particular relation instance. Such patterns do not generalize well, and are unlikely to be very productive when applied to novel text.

4.2 Dataset

We apply the pattern extraction approach described in Section 2 to create a dataset for 25 relations from the domains *awards*, *business* and *personal relationships*. We use Freebase as our knowledge base, and retrieve 200K relation instances as seed knowledge. We then create a text corpus by querying Bing with the seeds as input, and retrieving the top 100 results per query. From these documents, we extract more than 3M sentences mentioning a seed relation instance. The resulting pattern dataset contains 1.5M unique patterns. Since a manual evaluation of all these patterns would be too resource-intensive, we select a subset based on the pattern filtering algorithm proposed by Moro et al. (2013).

We then sample a small set of sentences $(3-5)$ for each pattern, and conduct an initial pass over the data with human annotators that judge whether these sentences express the target relation or not. We discard all patterns whose sentences do not express the relation. The final dataset for manual evaluation consists of more than 8K patterns with all their source sentences.

5 Pattern observations

Three annotators evaluated 1185 patterns for the relations *award honor* (510 patterns), *acquisition* (224) and *marriage* (451), using the guidelines described in the previous section. Each annotator evaluated the patterns of a single relation.[3]

5.1 Error classes

The annotators identified six main error classes, which are listed in Table 1. Three of the classes relate to preprocessing errors (*PIPE-S*, *PIPE-NER*, *PIPE-PT*), the other three encompass semantic mistakes in patterns or source sentences (*NEX-P*, *NEX-S*, *IMP-S*).

The error class *PIPE-S* is used for ungrammatical sentences and patterns resulting from sentence boundary detection errors. In example (1) in Table 1, the category label tokens "Personal life" are interpreted as relevant elements of the extracted pattern. *PIPE-NER* errors refer to patterns with arguments that are semantically or grammatically incongruent with the ones tagged in the sentence, as well as entity type errors. In example (2), the title of the book has not been recognized as an entity, and the lemmas "leave" and "us" are included as lexical elements in the pattern. The category *PIPE-PT* is applied to patterns derived from defective dependency parse trees. In example (3),

[3]We used a separate relation, *siblings*, to establish a shared set of evaluation principles among the annotators. In future work, we plan to have multiple annotations per pattern, e.g., to analyze inter-annotator agreement.

#	Error class	Description	Example
1	PIPE-S	Sentence segmentation error	Personal life On July 5, 2003, Banks married sportswriter and producer Max Handelman, who had been her boyfriend since she met him on her first day at college, September 6, 1992. *(marriage)*
2	PIPE-NER	NER tagging error	Rahna Reiko Rizzuto is the author of the novel, Why She Left Us, which won an American Book Award in 2000. *(award honor)*
3	PIPE-PT	Dependency parsing error	*Say won a Caldecott Medal for his illustrations in Grandfather's Journey. *(award honor)*
4	NEX-P	Relation is not expressed in pattern	Julian joined Old Mutual in August 2000 as Group Finance Director, moving on to become CEO of Skandia following its purchase by Old Mutual in February 2006. *(acquisition)*
5	NEX-S	Relation is not expressed in text	The 69th Annual Peabody Awards ceremony will be held on May 17 at the Waldorf-Astoria in New York City and will be hosted by Diane Sawyer, the award-winning anchor of ABCs World News. *(award honor)*
6	IMP-S	Relation is too implicit	The looming expiration of Lipitors patent in 2012 is a big reason Pfizer felt compelled to buy a company like Wyeth. *(acquisition)*

Table 1: Common error classes of dependency patterns for the relations *marriage*, *acquisition* and *award honor*. Underlined token sequences denote relation arguments, concepts with a dashed underline are additional pattern elements.

the parser interpreted the proper name *Say* as a finite verb.

The category *NEX-P* is used for dependency patterns that do not include any relation-relevant content words. In example (4), the most explicit word expressing an acquisition is the lemma "purchase". The pattern, however, extracts other parts of the source sentence. *NEX-S* applies to patterns that are based on sentences which do not express the relation of interest. In example (5), the target relation *award honor* is not expressed, instead, the host of the ceremony is erroneously identified as the winner of the prize. Finally, the category *IMP-S* marks patterns that are derived from sentences in which a relation is expressed merely implicitly. Judging from the source sentence in example (6), we cannot be entirely sure whether or not an acquisition took place because "felt compelled to" might only express a momentary mindset of the company's leaders that was not followed by action.

5.2 Pattern statistics

Table 2 summarizes the distribution of correct and incorrect dependency patterns for the three relations *marriage*, *award honor* and *acquisition*. We find that between 24% and 61% of the learned dependency patterns are defective, between 21% and 55% are labeled as correct. For the relation *acquisition*, more than 60% of the patterns are labeled as "INCORRECT", which is much higher than for the other two relations. "CORRECT, BUT TOO SPECIFIC" patterns make up between 12% and 30% of the total number of patterns.

	award honor	*acquisition*	*marriage*
Correct	54.7%	21.0%	40.0%
Correct, but too specific	12.4%	14.7%	29.9%
Incorrect	24.3%	60.7%	24.6%
Uncertain	8.6%	3.6%	5.5%

Table 2: Distribution of pattern categories

Table 3 gives details on the distribution of the error classes for the same relations. The two predominant error classes are PIPE-NER and NEX-S. The distribution of error classes varies significantly between the different relations. *PIPE-NER* is the category most frequently found in *award honor*. Sentences in this category often mention the titles of works the prize was awarded for. If those titles are not recognized as entities by the NER tagger, the dependency parsing fails and parts of the title can erroneously end up in the pattern. For the *acquisition* relation, the vast majority of errors can be assigned to the category *NEX-S*. In these cases, a relation between two or more or-

	award honor	*acquisition*	*marriage*
PIPE-S	1	2	11
PIPE-NER	78	2	10
PIPE-PT	33	4	14
NEX-P	3	19	26
NEX-S	2	107	2
IMP-S	5	1	34
Other	2	1	13

Table 3: Distribution of error classes

ganizations is often expressed in the source sentences, e.g., that "company *X* is a subsidiary of company *Y*", but no statement is made about the act of purchase. For the *marriage* relation, the most frequent error type was *IMP-S*, mainly resulting from sentences stating a divorce, which we do not consider as explicit mentions of the *marriage* relation. A final observation that can be made from Table 3 is that 42% of the errors are preprocessing pipeline errors.

6 Conclusions and future work

We presented *PatternJudge*, a linguistic annotation tool for manual evaluation of dependency patterns. The tool allows human experts to inspect dependency patterns and their associated source sentences, to categorize patterns, and to identify error classes. The annotated patterns can be used to create datasets that enable machine learning approaches to pattern quality estimation and relation extraction. We showed how the tool can be used to perform a pattern error analysis on three semantic relations. Our study indicates that textual entailment may play an important role for relation extraction, since many relations are not expressed explicitly in texts. We also observe that close interactions among semantically similar relations should be reflected in the pattern discovery approach. In future work, we will extend the *PatternJudge* tool to provide a better interface for defining and assigning error classes. In addition, our annotators are currently evaluating the pattern dataset for a larger set of semantic relations, which will allow us to extend the initial study presented in this work.

Acknowledgments

This research was partially supported by the German Federal Ministry of Education and Research (BMBF) through the projects ALL SIDES (01IW14002) and BBDC (01IS14013E), by the German Federal Ministry of Economics and Energy (BMWi) through the project SDW (01MD15010A), and by Google through a Focused Research Award granted in July 2013.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proc. of SIGMOD*, pages 1247–1250.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. The Automatic Content Extraction (ACE) Program - Tasks, Data, and Evaluation. In *Proc. of LREC*.

Ralph Grishman. 2012. Information Extraction: Capabilities and Challenges. Technical report, NYU Dept. CS.

Sebastian Krause, Hong Li, Hans Uszkoreit, and Feiyu Xu. 2012. Large-Scale Learning of Relation-Extraction Rules with Distant Supervision from the Web. In *Proc. of ISWC*, pages 263–278.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant Supervision for Relation Extraction Without Labeled Data. In *Proc. of ACL-IJCNLP*, pages 1003–1011.

Andrea Moro, Hong Li, Sebastian Krause, Feiyu Xu, Roberto Navigli, and Hans Uszkoreit. 2013. Semantic Rule Filtering for Web-Scale Relation Extraction. In *Proc. of ISWC*, pages 347–362.

Alan Ritter, Luke Zettlemoyer, Mausam, and Oren Etzioni. 2013. Modeling Missing Data in Distant Supervision for Information Extraction. *TACL*, 1:367–378.

Mark Stevenson and Mark Greenwood. 2005. A semantic approach to IE pattern induction. In *Proc. of ACL*, pages 379–386.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance Multi-label Learning for Relation Extraction. In *Proc. of EMNLP*, pages 455–465.

Hans Uszkoreit and Feiyu Xu. 2013. From Strings to Things – Sar-Graphs: A New Type of Resource for Connecting Knowledge and Language. In *Proc. of WS on NLP and DBpedia*.

Feiyu Xu, Hans Uszkoreit, and Hong Li. 2007. A Seed-driven Bottom-up Machine Learning Framework for Extracting Relations of Various Complexity. In *Proc. of ACL*, pages 584–591.