Unsupervised Parsing for Generating Surface-Based Relation Extraction Patterns

Jens Illig
University of Kassel
Wilhelmshöher Allee 73
D-34121 Kassel, Germany
illig@cs.uni-kassel.de

Benjamin Roth and **Dietrich Klakow**
Saarland University
D-66123 Saarbrücken, Germany
{benjamin.roth, dietrich.klakow}
@lsv.uni-saarland.de

Abstract

Finding the right features and patterns for identifying relations in natural language is one of the most pressing research questions for relation extraction. In this paper, we compare patterns based on supervised and unsupervised syntactic parsing and present a simple method for extracting surface patterns from a parsed training set. Results show that the use of surface-based patterns not only increases extraction speed, but also improves the quality of the extracted relations. We find that, in this setting, unsupervised parsing, besides requiring less resources, compares favorably in terms of extraction quality.

1 Introduction

Relation extraction is the task of automatically detecting occurrences of expressed relations between entities in a text and structuring the detected information in a tabularized form. In natural language, there are infinitely many ways to creatively express a set of semantic relations in accordance to the syntax of the language. Languages vary across domains and change over time. It is therefore impossible to statically capture all ways of expressing a relation.

Most relation extraction systems (Bunescu and Mooney, 2005; Snow et al., 2005; Zhang et al., 2006; Mintz et al., 2009; Alfonseca et al., 2012; Min et al., 2012) generalize semantic relations by taking into account statistics about the syntactic construction of sentences. Usually supervised parsers are applied for parsing sentences.

Statistics are then utilized to *machine-learn* how textual mentions of relations can be identified. Many researchers avoid the need for expensive corpora with manually labeled relations by applying a scheme called *distant supervision* (Mintz et al., 2009; Roth et al., 2013) which hypothesizes that all text fragments containing argument co-occurrences of known semantic relation facts indeed express these relations. Still, systems relying on supervised parsers require training from annotated treebanks, which are expensive to create, and highly domain- and language dependent when available.

An alternative is unsupervised parsing, which automatically induces grammars by structurally analyzing unlabeled corpora. Applying unsupervised parsing thus avoids the limitation to languages and domains for which annotated data is available. However, induced grammars do not match traditional linguistic grammars. In most of the research on parsing, unsupervised parsers are still evaluated based on their level of correspondence to treebanks. This is known to be problematic because there are several different ways of linguistically analyzing text, and treebank annotations also contain questionable analyses (Klein, 2005). Moreover, it is not guaranteed that the syntactic analysis which is most conforming to a general linguistic theory is also best suited in an extrinsic evaluation, such as for relation extraction.

In this work, we apply a supervised and an unsupervised parser to the relation extraction task by extracting statistically counted patterns from the resulting parses. By utilizing the performance of the overall relation extraction system as an indirect measure of a parser's practical qualities, we get a task-driven evaluation comparing supervised and unsupervised parsers. To the best of our knowledge, this is the first work to compare general-purpose unsupervised and supervised parsing on the application of relation extraction. Moreover, we introduce a simple method to obtain shallow patterns from syntactic analyses and show that, besides eliminating the need to parse text during system application, such patterns also increase extraction quality. We discover that, for this method, un-

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 100–105,
Gothenburg, Sweden, April 26-30 2014.

supervised parsing achieves better extraction quality than the more expensive supervised parsing.

1.1 Related Work

Unsupervised and weakly supervised training methods have been applied to relation extraction (Mintz et al., 2009; Banko et al., 2007; Yates and Etzioni, 2009) and similar applications such as semantic parsing (Poon and Domingos, 2009) and paraphrase acquisition (Lin and Pantel, 2001). However, in such systems, parsing is commonly applied as a separately trained subtask[1] for which supervision is used.

Hänig and Schierle (2009) have applied unsupervised parsing to a relation extraction task but their task-specific data prohibits supervised parsing for comparison.

Unsupervised parsing is traditionally only evaluated intrinsically by comparison to gold-standard parses. In contrast, Reichart and Rappoport (2009) count POS token sequences inside sub-phrases for measuring parsing consistency. But this count is not clearly related to application qualities.

2 Methodology

A complete relation extraction system consists of multiple components. Our system follows the architecture described by Roth et al. (2012). In short, the system retrieves queries in the form of entity names for which all relations captured by the system are to be returned. The entity names are expanded by alias-names extracted from Wikipedia link anchor texts. An information retrieval component retrieves documents containing either the name or one of the aliases. Further filtering retains only sentences where a named entity tagger labeled an occurrence of the queried entity as being of a suitable type and furthermore found a possible entity for the relation's second argument. For each candidate sentence, a classifier component then identifies whether one of the captured relation types is expressed and, if so, which one it is. Postprocessing then outputs the classified relation according to task-specific format requirements. Here, we focus on the relation type classifier.

2.1 Pattern Extraction

For our relation extraction system, we use a simple pattern matching framework. Whenever at least one candidate sentence containing two entities A and B matches one of the patterns extracted for a certain relation type R, the classifier states that R holds between A and B.

We experimented with two types of patterns. First, we simply parsed the training set and extracted shortest dependency path patterns. These patterns search for matches on the parse tree. Following Lin and Pantel (2001), the shortest path connecting two arguments in a dependency graph has been widely used as a representation of relation instance mentions. The general idea is that shortest paths skip over irrelevant optional parts of a sentence such as in *$1, who ... founded $2* where the shortest path pattern `$1←founded→$2` matches although an irrelevant relative clause appears between the arguments $1 and $2. Similar representations have been used by Mintz et al. (2009), Alfonseca et al. (2012) and Snow et al. (2005).

In a second set of experiments, we used the shortest dependency paths in parsed training sentences to generate surface-based patterns. These patterns search for matches directly on plain text and therefore do no longer rely on parsing at application time. The patterns are obtained by turning the shortest paths between relational arguments in the parsed training data into token sequences with gaps. The token sequences consist of all words in the sentence that appear on the shortest dependency path. Argument positions in the surface patterns are specified by special tokens `$1` and `$2`. At all places, where there are one or more tokens which are not on the shortest dependency path but which are surrounded either by tokens on the dependency path or by arguments, an asterisk represents up to four unspecified tokens. For the shortest path `$1←,←who→$2` connecting *Friedman* and *economist* in the DMV parse depicted in Figure 1, this method generates the pattern `$1, * $2 who`. As can be seen, such patterns can capture a conjunction of token presence conditions to the left, between, and to the right of the arguments. In cases where argument entities are not parsed as a single complete phrase, we generate patterns for each possible combination of outgoing edges from the two arguments. We dismiss patterns generated for less than four distinct argument entity pairs of

[1] An exception is the joint syntactic and semantic (supervised) parsing model inference by Henderson et al. (2013)

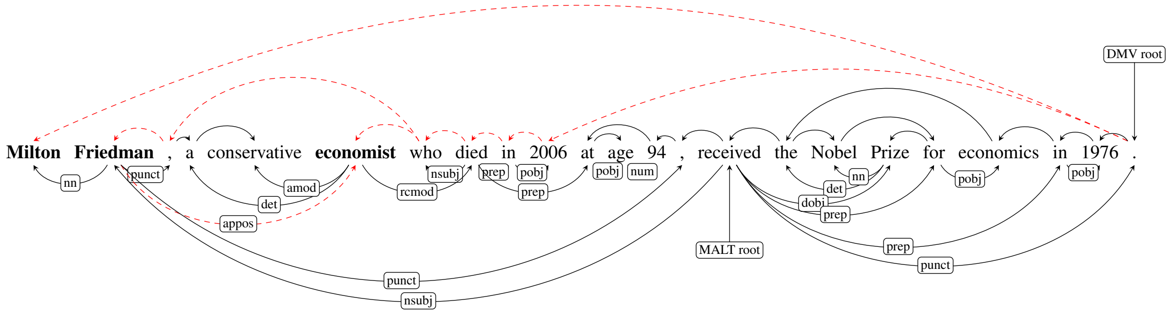
DMV root
Milton Friedman , a conservative economist who died in 2006 at age 94 , received the Nobel Prize for economics in 1976 .
nn
punct
amod
det
appos
nsubj
prep
pobj
rcmod
prep
pobj
num
punct
nsubj
MALT root
nn
det
dobj
prep
pobj
pobj
prep
punct

Figure 1: Comparison of a DMV (above text) and a MALT parse (below text) of the same sentence.

the same relation type. For each pattern, we calculate the precision on the training set and retain only patterns above a certain precision threshold.

2.2 Supervised and Unsupervised Parsing

Typical applications which require syntactic analyses make use of a parser that has been trained under supervision of a labeled corpus conforming to a linguistically engineered grammar. In contrast, unsupervised parsing induces a grammar from frequency structures in plain text.

Various algorithms for unsupervised parsing have been developed in the past decades. Headden (2012) gives a rather recent and extensive overview of unsupervised parsing models. For our work, we use the Dependency Model with Valence (DMV) by Klein and Manning (2004). Most of the more recent unsupervised dependency parsing research is based on this model. DMV is a generative head-outward parsing model which is trained by expectation maximization on part-of-speech (POS) sequences of the input sentences. Starting from a single root token, head tokens generate dependants by a probability conditioned on the direction (left/right) from the head and the head's token type. Each head node generates tokens until a stop event is generated with a probability dependent on the same criteria plus a flag whether some dependant token has already been generated in the same direction.

For comparison of unsupervised and supervised parsing, we apply the (Nivre, 2003) deterministic incremental parsing algorithm *Nivre arc-eager*, the default algorithm of the *MALT* framework[2] (Nivre et al., 2007). In this model, for each word token, an SVM classifier decides for a parser state transition, which, in conjunction with other decisions, determines where phrases begin and end.

3 Experiments

We used the plain text documents of the English Newswire and Web Text Documents provided for TAC KBP challenge 2011 (Ji et al., 2011). We automatically annotated relation type mentions in these documents by distant supervision using the online database Freebase[3], i.e. for all relation types of TAC KBP 2011, we took relation triples from Freebase and, applying preprocessing as described in Section 2, we retrieved sentences mentioning both arguments of some Freebase relation with matching predicted entity types. We hypothesize that all sentences express the respective Freebase relation. This way we retrieved a distantly supervised training set of 480 622 English sentences containing 92468 distinct relation instances instantiating 41 TAC KBP relation types.

3.1 Training and Evaluation

From our retrieved set of sentences, we took those with a maximum length of 10 tokens and transformed them to POS sequences. We trained DMV only on this dataset of short POS sequences, which we expect to form mentions of a modeled relation. Therefore, we suspect that DMV training assigns an increased amount of probability mass to dependency paths along structures which are truly related to these relations. We used the DMV implementation from Cohen and Smith (2009) [4].

For the supervised *Nivre arc-eager* parser we used *MALT* (Nivre et al., 2007) with a pre-trained Penn Treebank (Marcus et al., 1993) model[5]. As a baseline, we tested *left branching* parses i.e.

[2] `http://www.maltparser.org` as of Nov. 2013

[3] `http://www.freebase.com` as of Nov. 2013

[4] publicly available at `http://www.ark.cs.cmu.edu/DAGEEM/` as of Nov. 2013 (parser version 1.0).

[5] `http://www.maltparser.org/mco/english_parser/engmalt.linear-1.7.mco` as of Nov. 2013

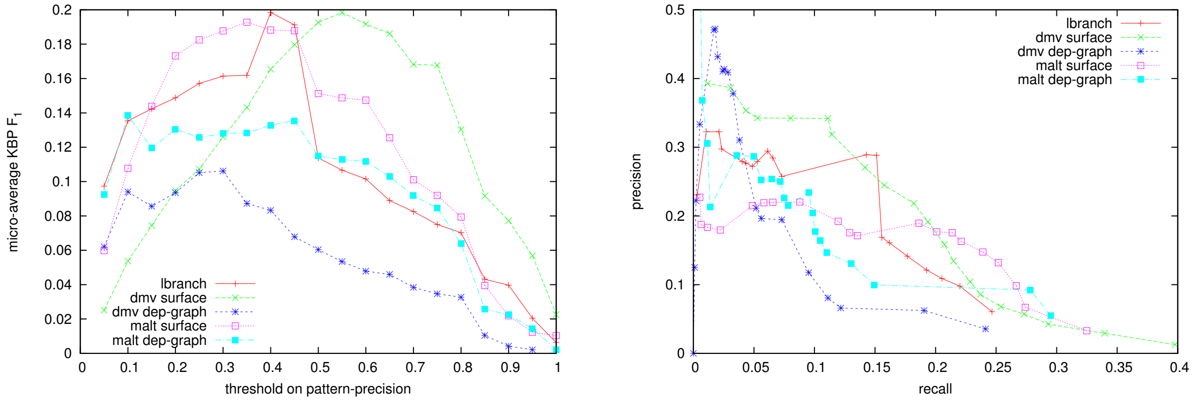

Figure 2: micro-averaged F_1 and precision&recall results for varied training precision thresholds

pattern set (+additional DMV pattern)		precision	recall	F_1
MALT generated patterns only		**.1769**	**.2010**	**.1882**
+p:title	$1 * $2 of	+0.73%	+8.40%	+4.14%
+p:title	$1 , * $2 of	+0.90%	+4.22%	+2.39%
+o:state_of_hqs	$1 * in * , $2	+1.35%	+1.59%	+1.43%
+p:title	$1 , * $2 who	+0.90%	+1.35%	+1.22%
+o:parents	$1 , * by $2	+0.62%	+1.35%	+1.06%
+o:city_of_hqs	$1 , * in $2 ,	+1.01%	+1.04%	+1.00%
+p:origin	$2 's $1 won the	+0.84%	+1.04%	+0.95%
+p:employee_of	$1 * $2 's chief	+0.28%	+1.04%	+0.79%
+o:website	$1 : $2	+0.28%	+1.04%	+0.79%

Table 1: DMV patterns improving MALT results the most, when added to the MALT patternset

dependency trees solely consisting of head-to-dependent edges from the right to the left[6].

All the extracted sentences were parsed and patterns were extracted from the parses. The patterns were then applied to the corpus and their precision was determined according to Freebase. With different cut-off values on training precision, the full relation extraction pipeline described in Section 2 was evaluated with respect to the Slot Filling test queries of TAC KBP 2011.

3.2 Results

Figure 2 (left) depicts F_1-measured testset results for pattern sets with varying training precision thresholds. Figure 2 (right) shows a precision recall plot of the same data points.

As can be seen in Figure 2 (left), flattening graph patterns to surface-based patterns increased the overall F_1 score. The curve for MALT generated surface patterns in Figure 2 (right) shows no increase in precision towards low recall levels where only the highest-training-precision patterns are retained. This indicates a lack of precision in MALT-based surface patterns. In contrast, the corresponding DMV-based graph increases monotonically towards lower recall levels, which is reflected by the highest F_1 score (Figure 2, left).

Table 1 shows the increases in evaluation score of those DMV-generated patterns which help most to more precisely identify relations when added to the set of all MALT-generated patterns (sorted by F_1 score). Figure 1 compares the syntactic analyses of MALT and DMV for an example sentence where DMV generates one of the listed patterns. The numbers of Table 1 indicate that such patterns are missing without alternatives in the pattern set gained from supervised parsing.

4 Conclusion

We have presented a simple method for generating surface-based patterns from parse trees which, besides avoiding the need for parsing test data, also increases extraction quality. By comparing supervised and unsupervised parsing, we furthermore found that unsupervised parsing not only eliminates the dependency on expensive domain-specific training data, but also produce surface-based extraction patterns of increased quality. Our results emphasize the need for task-driven evaluation of unsupervised parsing methods and show that there exist indicative structures for relation extraction beyond widely agreed-on linguistic syntax analyses.

5 Acknowledgements

Benjamin Roth is a recipient of the Google Europe Fellowship in Natural Language Processing, and this research is supported in part by this Google Fellowship.

[6] Since for such parses the shortest path is the complete observed word sequence between the two relation arguments, surface and parse-tree patterns become equal.

References

Enrique Alfonseca, Katja Filippova, Jean-Yves Delort, and Guillermo Garrido. 2012. Pattern learning for relation extraction with a hierarchical topic model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 54–59, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shay B. Cohen and Noah A. Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 74–82, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christian Hänig and Martin Schierle. 2009. Relation extraction based on unsupervised syntactic parsing. In Gerhard Heyer, editor, *Text Mining Services*, Leipziger Beiträge zur Informatik, pages 65–70, Leipzig, Germany. Leipzig University.

William Headden. 2012. *Unsupervised Bayesian Lexicalized Dependency Grammar Induction*. Ph.D. thesis, Brown University.

James Henderson, Paola Merlo, Ivan Titov, and Gabriele Musillo. 2013. Multi-lingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational Linguistics*, 39(4).

Heng Ji, Ralph Grishman, and Hoa Dang. 2011. Overview of the TAC2011 knowledge base population track. In *TAC 2011 Proceedings Papers*.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. In *ACL*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Klein. 2005. *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis, Stanford University.

Dekang Lin and Patrick Pantel. 2001. DIRT: Discovery of Inference Rules from Text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 323–328, New York, NY, USA. ACM Press.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, 19(2):313–330, June.

Bonan Min, Xiang Li, Ralph Grishman, and Sun Ang. 2012. New york university 2012 system for kbp slot filling. In *Proceedings of the Fifth Text Analysis Conference (TAC 2012)*. National Institute of Standards and Technology (NIST), November.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Roi Reichart and Ari Rappoport. 2009. Automatic selection of high quality parses created by a fully unsupervised parser. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 156–164, Stroudsburg, PA, USA. Association for Computational Linguistics.

Benjamin Roth, Grzegorz Chrupala, Michael Wiegand, Singh Mittul, and Klakow Dietrich. 2012. Generalizing from freebase and patterns using cluster-based distant supervision for tac kbp slotfilling 2012. In *Proceedings of the Fifth Text Analysis Conference (TAC 2012)*, Gaithersburg, Maryland, USA, November. National Institute of Standards and Technology (NIST).

Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. 2013. A survey of noise reduction methods for distant supervision. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 73–78. ACM.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, Cambridge, MA.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *J. Artif. Int. Res.*, 34(1):255–296, March.

Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 825–832, Stroudsburg, PA, USA. Association for Computational Linguistics.