Text Rewriting Improves Semantic Role Labeling (Extended Abstract) *

Kristian Woodsend and **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
kristian@woodsend.me.uk mlap@inf.ed.ac.uk

Abstract

Large-scale annotated corpora are a prerequisite to developing high-performance NLP systems. Such corpora are expensive to produce, limited in size, often demanding linguistic expertise. In this paper we use text rewriting as a means of increasing the amount of labeled data available for model training. Our method uses automatically extracted rewrite rules from comparable corpora and bitexts to generate multiple versions of sentences annotated with gold standard labels. We apply this idea to semantic role labeling and show that a model trained on rewritten data outperforms the state of the art on the CoNLL-2009 benchmark dataset.

1 Introduction

Recent years have witnessed increased interest in the automatic identification and labeling of the *semantic roles* conveyed by sentential constituents [Gildea and Jurafsky, 2002]. The goal of the semantic role labeling task is to discover the relations that hold between a predicate and its arguments in a given input sentence (e.g., "who" did "what" to "whom", "when", "where", and "how").

(1) [Mrs. Yeargin]$_{\mathrm{A0}}$ [gave]$_{\mathrm{V}}$ [the questions and answers]$_{\mathrm{A1}}$ [two days before the examination]$_{\mathrm{TMP}}$ to [two low-ability geography classes]$_{\mathrm{ARG2}}$.

In sentence (1), A0 represents the *Agent* or *giver*, A1 represents the *theme* or *thing given*, A2 represents the *Recipient*, TMP is a *temporal modifier* indicating when the action took place, and V determines the boundaries of the predicate. The semantic roles in the example are labeled in the style of PropBank [Palmer *et al.*, 2005], a broad-coverage human-annotated corpus of semantic roles and their syntactic realizations. Under the PropBank annotation framework each predicate is associated with a set of core roles (named A0, A1, A2, and so on) whose interpretations are specific to that predicate and a set of adjunct roles such as *location* or *time* whose interpretation is common across predicates (e.g., *two days before the examination* in sentence (1) above).

This type of semantic information is shallow but relatively straightforward to infer automatically and useful for the development of broad coverage, domain-independent language understanding systems. Indeed, the analysis produced by existing semantic role labelers has been shown to benefit a wide spectrum of applications ranging from information extraction [Surdeanu *et al.*, 2003] and question answering [Shen and Lapata, 2007], to machine translation [Wu and Fung, 2009] and summarization [Melli *et al.*, 2005].

Most SRL systems to date conceptualize the semantic role labeling task as a supervised learning problem and rely on role-annotated data for model training. Supervised methods deliver reasonably good performance, with F1-scores in the low eighties on standard test collections for English. They rely primarily on syntactic features (such as path features) in order to identify and classify roles. This has been a mixed blessing as the path from an argument to the predicate can be very informative but also quite complicated. Many paths through the parse tree are likely to occur a relatively small number of times (or not at all) resulting in very sparse information for the classifier to learn from. Even if the training data includes examples for a specific predicate and set of arguments, unless a test sentence contains them in the same syntactic structure, then as far as the classifier is concerned, the labeling of items within the two sentences is unrelated.

Our idea is to use rewrite rules in order to create several syntactic variants for a sentence, thus alleviating the training requirements for semantic role labeling. Rewrite rules are typically synchronous grammar rules defining how a sequence of *source* terminals and nonterminals rewrites to a sequence of *target* terminals and nonterminals. Such rules are most often extracted from *monolingual* corpora containing parallel translations of the same source text [Barzilay and McKeown, 2001; Pang *et al.*, 2003], *bilingual* corpora consisting of documents and their translations [Bannard and Callison-Burch, 2005; Callison-Burch, 2007], or *comparable* corpora such as Wikipedia revision histories [Coster and Kauchak, 2011; Woodsend and Lapata, 2011]. Examples of rewrites are given in Table 1. These include transforming passive to active sentences (see sentence pair (1) in Table 1), splitting a long and complicated sentence into several shorter ones (2), removing redundant parts of a sentence (3), reordering parts in a sentence (4), deleting appositives (5), transforming a prepositional phrase into a genitive (6), and so on.

*This paper is an extended abstract of an article in the Journal of Artificial Intelligence Research [Woodsend and Lapata, 2014].

Source	Target
1. The retreating guerrillas were soon pursued by the government forces.	Government forces soon pursued the retreating guerrillas.
2. A survey conducted by the Gallup Poll last summer indicated that one in four Americans takes cues from the stars or believes in ghosts.	A survey was conducted by the Gallup Poll last summer. It indicated that one in four Americans takes cues from the stars or believes in ghosts.
3. The examiner who was kind let the student finish his lunch.	The kind examiner let the student finish his lunch.
4. Because she didn't know the rules, she died.	She died, because she didn't know the rules.
5. Mexico City, the biggest city in the world, has many interesting archaeological sites.	Mexico City has many interesting archaeological sites.
6. The arrival of the train was unexpected.	The train's arrival was unexpected.

Table 1: Examples of syntactic rewriting.

We automatically extract syntactic rewrite rules from corpora and use them to generate multiple versions of role annotated sentences whilst preserving their original semantic roles. We therefore expand the training data with a wide range of syntactic variations for each predicate-argument combination and then learn a semantic role labeler on the expanded dataset. The approach we describe essentially increases the size of the training data by creating many different syntactic variations for different predicates and their roles.

Using the CoNLL-2009 benchmark dataset and the best scoring system [Björkelund *et al.*, 2009], we show experimentally, that syntactic transformations improve SRL performance beyond the state of the art. Importantly, our approach can be used in combination with any SRL learner or role-annotated data.

2 Method

We describe the general idea behind our algorithm and then move on to present our specific implementation. We define a *transformation* to be a function that maps an example sentence s into a modified sentence s'. Suppose now that there are labels associated with example s. In the context of this paper, these are semantic role labels. Labels could be defined over spans of tokens, but here we use the CoNLL 2008–9 formalism where it is the head word of the span that is labelled. The transformation function is therefore a mapping between tokens t in sentence s to tokens t' in s'. We do not require that the mapping involves all the tokens of s or s', but we do require that the mappings are one-to-one.

A *label-preserving transformation* is a mapping from (some of the) tokens t in example s to tokens t' in s', such that the (correct) labels of t' are identical to the labels of its source tokens t for all the token mappings defined in the transformation. In other words, those labels that could be preserved, have been preserved, and no others have been introduced.

Our approach boils down to three steps: (a) extracting transformations, (b) refining transformations, and (c) generating and labeling an extended corpus.

2.1 Extracting Transformations

A standard gold annotated corpus is used to train an initial semantic role labeling model. Meanwhile, a set of candidate transformations are extracted from some suitable comparable or parallel corpus. This full set of transformations is used to rewrite the gold corpus, creating a much extended corpus which inevitably will contain grammatically or semantically incorrect sentences. The extended corpus is next automatically labeled using the original SRL model after preprocessing through a normal SRL pipeline, without knowledge of the transformation functions involved.

Conceptually a wide range of text-rewriting transformation functions could be included, such as paraphrasing, simplification or translation into another language. Here, we focus on transformation functions that can be expressed in synchronous context-free grammars [Aho and Ullman, 1969]. Synchronous rules operate on parse tree constituents in a context-free manner, and typically modify the syntax. The transformations we consider can be sub-categorized into:

1. *Statement extraction.* Constituents of a sub-tree of the parse tree are identified, extracted from their context and rewritten as a complete sentence, typically shorter and simpler, although not necessarily so.
2. *Compression.* The original sentence is rewritten by compressing constituents of the parse tree, typically by deleting nodes.
3. *Insertion.* New elements are added to the parse tree. As significant chunks of new text would have semantic role information of their own, in practice these insertions are often additional punctuation to clarify the scope of phrases, or a simple structure such as "It is ... ." to aid in statement extraction.
4. *Substitution.* Through a lexicalized synchronous grammar, text can be replaced with new text, and paraphrases represented.

We obtain a set of possible transformations from monolingual comparable corpora drawn from Wikipedia and bitexts.

In a synchronous tree-substitution grammar (STSG), rules specify how to map tree fragments of the source parse tree into fragments in the target tree, recursively and free of context. In our experiments, we investigate two STSG variants, the strictly synchronous tree substitution grammar T3 [Cohn and Lapata, 2009], which was originally developed for the task of text compression, but does support a full range of transformation operations; and the quasi-synchronous tree

Grammar	Examples	Type	Label
Original	Bell, based in Los Angeles, makes and distributes electronic, computer and building products.		
T3	Bell, based in Los Angeles, makes and distributes. ⟨NP, NP⟩ → ⟨[NP ADJP$_{\boxed{\epsilon}}$ NNS$_{\boxed{\epsilon}}$], [NP]⟩	Comp	+
QTSG	Bell was based in Los Angeles. ⟨NP, S⟩ → ⟨[NP NP$_{\boxed{1}}$, VP$_{\boxed{2}}$,], [S NP$_{\boxed{1}}$ [VP [VBD was] VP$_{\boxed{2}}$] .]⟩	Ext	+
PPDB	Bell, founded in Los Angeles, makes and distributes electronic, computer and building products. ⟨VP, VP⟩ → ⟨[VP [X based] PP$_{\boxed{1}}$], [VP [X founded] PP$_{\boxed{1}}$]⟩	Sub	–
H&S	Bell makes. Bell distributes. Bell is based in Los Angeles.	Ext	+

Table 2: Examples of transformation rules extracted using T3, QTSG and PPDB grammar formalisms, applied to the sentence marked Original. The *type* column indicates whether the rule is statement extraction (Ext), compression (Comp), insertion (Ins) or substitution (Sub). The symbols +/– in the *label* column indicate whether the sample was classified as positive (i.e., argument label preserving) and forms part of extended training corpus, or not. Boxed indices are short-hand notation for the alignment, $\sim$.

substitution grammar QTSG [Woodsend and Lapata, 2011], which has been used in text simplification and summarization [Woodsend and Lapata, 2012].

We also obtain transformation rules from the ParaPhrase DataBase (PPDB), a collection of English (and Spanish) paraphrases derived from large bilingual parallel corpora [Ganitkevitch *et al.*, 2013]. A variety of paraphrases (lexical, phrasal, and syntactic) are obtained through bilingual pivoting [Bannard and Callison-Burch, 2005].

Our experiments primarily make use of automatically learned transformations since these can be adapted to different tasks, domains or languages. However, for the proposed approach it is not necessary that transformation functions are acquired automatically — such functions could be also crafted by hand. We thus also investigated the effectiveness of rewrites generated by the system of Heilman and Smith [2010] (henceforth H&S), which uses a sophisticated handcrafted rule-based algorithm to extract simplified declarative sentences in English from syntactically complex ones.

Table 2 shows examples of rules extracted using the T3, QTSG and PPDB grammar formalisms applied to a sentence from the CoNLL dataset. The *type* column of Table 2 indicates whether the transformation could be classed as statement extraction, compression, insertion, or substitution. As reflected in the table, T3 captures compression transformations by deleting nodes in the parse tree; QTSG rules are a range of mainly syntactic transformations; and PPDB transformations are substitutions of words or short phrases.

2.2 Refining Transformations

We could in theory use this extended corpus as the basis of training a further SRL model. However, it will contain many errors, and is unlikely to yield useful information to guide the model. One approach could be to manually correct the rewrites that have been generated automatically, but this would be very time and resource-intensive. Instead, we do the corrections automatically, and create an extended corpus where the rewrites do not impair the quality of the training data. We therefore learn which rules yield accurate rewrites, i.e., rewrites which preserve the labels of the gold-standard. Our intuition is that, given a large number of possible rewrites, the SRL model will in general label the accurate rewrites correctly and mis-label the erroneous sentences, due to it finding them more confusing. We thus compare the semantic role labels produced by the model with the labels for corresponding predicate-argument pairs in the gold corpus, and provide them as samples to train a binary classifier (here an SVM) which learns to predict which rewrites are likely to be successful and which are problematic.

Each rewritten sentence is classed as a positive sample if the SRL model predicts the same labels for the transformed sentence as those it predicted for the original, or the labels have now been corrected with respect to the gold labels. If, however, a semantic role is no longer predicted correctly, or missed, or an erroneous role introduced, this is classified as a negative sample, as such a sample is likely to harm the training of a new SRL model. To capture the full impact of a candidate transformation function, a sentence is provided as a positive sample to the classifier only if all the labels (i.e., all predicates and arguments) from the source sentence have been successfully projected onto the rewrite. Referring again to Table 2, the final column indicates whether these example rewrites were positive or negative. Note that no refining was used on the H&S outputs.

2.3 Generating the Extended Corpus

Once the SVM has identified the refined set of transformation functions, these transformations are used to create an extended training corpus. This time, knowledge of the transformation function is involved to project the labels that correspond to the original gold corpus. In the case of SRL, the labels describe the predicate and its arguments. This extended corpus supplements the original gold standard corpus, and the combination is then used to create a further SRL model.

It is worth noting that our method does not impinge on the actual process of learning an SRL model, as it is concerned with the preparation of training data. We therefore believe it can be applied to a range of SRL modeling approaches, and that gains in performance we achieve are largely orthogonal to those that could be made by improving other aspects of the learning process.

QTSG	⟨NP, NP⟩	→	⟨[NP NP$_{\boxed{1}}$, NP$_{\boxed{\epsilon}}$ CC NP$_{\boxed{\epsilon}}$], [NP NP$_{\boxed{1}}$]⟩
	⟨NP, S⟩	→	⟨[NP NP$_{\boxed{1}}$ PP$_{\boxed{2}}$], [S *It is* NP$_{\boxed{1}}$ PP$_{\boxed{2}}$.]⟩
PPDB	⟨ADJP, ADJP⟩	→	⟨[ADJP *just as* JJ$_{\boxed{1}}$], [ADJP *equally* JJ$_{\boxed{1}}$]⟩
	⟨PP, PP⟩	→	⟨[PP *in the past month*], [PP *in the last month*]⟩

Table 3: Examples of QTSG and PPDB synchronous grammar rules given high importance during refinement.

	In-domain F1	*Out-of-domain* F1
Original	80.41	68.40
H&S	80.70 †	67.75
PPDB	81.10 †‡	68.80 †‡
T3	81.05 †‡	68.90 ‡
QTSG	81.09 †‡	69.62 †‡
PPDB+T3	81.09 †‡	68.95 †‡
PPDB+QTSG	81.29 †‡	69.71 †‡
T3+QTSG	81.23 †‡	69.49 †‡
PPDB+QTSG+T3	81.37 †‡	69.74 †‡

Table 4: Performance in the labeling of semantic arguments (predicate word sense information removed). † Difference from Original is significant at $p < 0.01$. ‡ Difference from H&S is significant at $p < 0.01$.

3 Experimental Results

Our experiments were primarily designed to answer the following questions. Does text rewriting generally improve SRL performance? Does it matter which transformation rules to use, i.e., are some rules better than others? Are the transformation rules useful on out-of-domain data?

Transformation rules improve F1 across the board

For the training corpora rewritten by the H&S system, the T3, QTSG, and PPDB grammars, all of the resulting SRL models significantly ($p < 0.01$) improve over a model trained on the original corpus in the task of SRL performance on the in-domain CoNLL-2009 test set. Recall shows the largest increase, particularly with the acquired synchronous grammars, indicating that the increased training data is resulting in better coverage. Generally this is not at the expense of precision which in all cases apart from PPDB has increased as well. Significant gains are also seen in the acquired grammars compared to the H&S system, with the exception of T3 where there is greater variation in its performance.

Transformation rules improve semantic role assignment for verbal and nominal predicates

An interesting result is that much of the gain in performance seen here by rewriting the training corpus comes through improving semantic role assignment (Table 4). It appears that introducing syntactic variation in the training data provides the model with wider coverage in syntactic dependency paths between predicate and arguments.

Transformation rules improve performance of relations involving long dependency paths

The *dependency path* (the sequence of arcs through the syntactic dependency tree) between a predicate and its argument is typically short. Existing SRL models are highly accurate over single arcs—the original SRL model has an F1-score of almost 89%—but prediction accuracy drops considerably as the dependency path grows. Adding rewrites to the training set improves prediction accuracy for almost all combinations of transformation grammar and dependency path distance, and the largest gains are seen when the number of arcs in the dependency path is between three and six. Improvements in F1-score are observed for individual grammars and their combination (PPDB+QTSG+T3).

Transformation rules improve performance even when a global reranker is used

The SRL system we used [Björkelund *et al.*, 2009] can optionally incorporate a global reranker [Toutanova *et al.*, 2005]. The reranker re-scores the complete predicate-argument structure, using features from all stages of the local pipeline and additional features representing the sequence of core argument labels for the current predicate. Training on the extended data gives further increases in performance, thouugh these are now smaller. This indicates that the global reranker is compensating for some, but not all, of the new information contained in the extended training data.

4 Conclusions

In this paper we investigated the potential of text rewriting as a means of increasing the amount of training data available for supervised NLP tasks. Our method automatically extracts rewrite rules from comparable corpora and uses them to generate multiple syntactic variants for sentences annotated with gold standard labels. Application of our method to semantic role labeling reveals that syntactic transformations improve SRL performance beyond the state of the art on the CoNLL 2009 benchmark dataset. Specifically, we experimentally show that (a) rewrite rules, whether automatic or hand-written, consistently improve SRL performance, although automatic variants tend to perform best; (b) syntactic transformations improve SRL performance both within- and out-of-domain; and (c) improvements are observed across learners, even when using a global reranker.

Acknowledgments

We acknowledge the financial support of EPSRC (EP/K017845/1) in the framework of the CHIST-ERA READERS project.

References

[Aho and Ullman, 1969] Alfred V. Aho and Jeffrey D. Ullman. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3(1):37–56, 1969.

[Bannard and Callison-Burch, 2005] Colin Bannard and Chris Callison-Burch. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of the 43rd ACL*, pages 597–604, Ann Arbor, MI, 2005.

[Barzilay and McKeown, 2001] Regina Barzilay and Kathy McKeown. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of the ACL/EACL*, pages 50–57, Toulouse, France, 2001.

[Björkelund *et al.*, 2009] Anders Björkelund, Love Hafdell, and Pierre Nugues. Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 43–48, Boulder, Colorado, June 2009. Software retrieved from `https://code.google.com/p/mate-tools/`.

[Callison-Burch, 2007] Chris Callison-Burch. *Paraphrasing and Translation*. PhD thesis, University of Edinburgh, 2007.

[Cohn and Lapata, 2009] Trevor Cohn and Mirella Lapata. Sentence Compression as Tree Transduction. *Journal of Artificial Intelligence Research*, 34:637–674, 2009.

[Coster and Kauchak, 2011] William Coster and David Kauchak. Simple English Wikipedia: A New Text Simplification Task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA, June 2011.

[Ganitkevitch *et al.*, 2013] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia, June 2013. We used the prepackaged "small" constituent syntactic subset of PPDB, retrieved from `http://paraphrase.org`.

[Gildea and Jurafsky, 2002] Daniel Gildea and Daniel Jurafsky. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245–288, 2002.

[Heilman and Smith, 2010] Michael Heilman and Noah Smith. Extracting Simplified Statements for Factual Question Generation. In *Proceedings of the 3rd Workshop on Question Generation*, pages 11–20, Carnegie Mellon University, PA, 2010. Software available at `http://www.ark.cs.cmu.edu/mheilman/questions/`.

[Melli *et al.*, 2005] Gabor Melli, Yang Wang, Yudong Liu, Mehdi M. Kashani, Zhongmin Shi, Baohua Gu, Anoop Sarkar, and Fred Popowich. Description of SQUASH, the SFU Question Answering Summary Handler for the DUC-2005 Summarization Task. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing Document Understanding Workshop*, Vancouver, Canada, 2005.

[Palmer *et al.*, 2005] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106, 2005.

[Pang *et al.*, 2003] Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences. In *Proceedings of the NAACL*, pages 181–188, Edmonton, Canada, 2003.

[Shen and Lapata, 2007] Dan Shen and Mirella Lapata. Using Semantic Roles to Improve Question Answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 12–21, Prague, Czech Republic, 2007.

[Surdeanu *et al.*, 2003] Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 8–15, Sapporo, Japan, 2003.

[Toutanova *et al.*, 2005] Kristina Toutanova, Aria Haghighi, and Christopher Manning. Joint learning improves semantic role labeling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 589–596, Ann Arbor, Michigan, June 2005.

[Woodsend and Lapata, 2011] Kristian Woodsend and Mirella Lapata. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK., 2011. We used the *Wikipedia revisions* corpus, retrieved from `http://homepages.inf.ed.ac.uk/kwoodsen/wiki.html`.

[Woodsend and Lapata, 2012] Kristian Woodsend and Mirella Lapata. Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 233–243, Jeju Island, Korea, July 2012.

[Woodsend and Lapata, 2014] Kristian Woodsend and Mirella Lapata. Text rewriting improves semantic role labeling. *Journal of Artificial Intelligence Research*, 51:133–164, 2014.

[Wu and Fung, 2009] Dekai Wu and Pascale Fung. Semantic Roles for SMT: A Hybrid Two-Pass Model. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 13–16, Boulder, Colorado, 2009.