

Generalizable Features Help Semantic Role Labeling *

Li Yang

Linguistics Department, University of Michigan
Ann Arbor, Michigan, USA 48109
lyshane@umich.edu

Abstract. In this paper, we take on the challenge of developing effective generalizable features for the task of semantic role labeling in the constituency grammar framework. Based on the knowledge of argument structure, on the constraint imposed by context dependence defined in the theory of argument realization, and on the knowledge of moved and displaced core arguments, we design the base argument configuration (BAC) feature that generalizes across four types of syntactic structures involving moved and displaced core arguments. As part of the effort to derive this base argument configuration feature, we also identify the core and non-core arguments in the system which is the first case in the field of semantic role labeling. Together with two levels of backoff features, the BAC feature effectively solve the argument classification task. However, as the experimental results show, our overall performance is affected by the argument identification module at present.

Keywords: SRL, semantic role labeling, core and non-core semantic arguments, context dependence, moved and displaced arguments

1 Introduction

The task of semantic role labeling identifies the semantic argument(s) of a verb and assigns a semantic role label to each semantic argument. Since the seminal work on semantic role labeling (SRL) by Gildea and Jurafsky (2000), the community has seen much effort dedicated to the task, which generated dozens of published SRL systems, including the ones participated in the shared tasks on CoNLL-2004, CoNLL-2005, and CoNLL-2008. In the past eight years, although researchers approached the SRL problem from different perspectives, they all focused on determining the appropriate syntactic/semantic knowledge and machine learning system to tackle the challenges in SRL (Carreras and Marquez, 2005; Surdeanu et al., 2008).

In terms of searching for the proper syntactic/semantic knowledge, the SRL researchers explored features based on two formalisms, namely constituency grammar and dependency grammar. The SRL systems constructed between the years of 2000 and 2006 investigated a variety of features that constituency grammar provides for. The main stream in this line of work had been developing specific features to cover diverse syntactic configurations that a predicate verb appears in (Toutanova et al., 2005; Haghighi et al., 2005), in addition to the generic syntactic features for the verb (Gildea and Jurafsky, 2000; Pradhan et al., 2004). As a result, few researchers investigated a feature engineering that could generalize across different syntactic configurations.

The lack of improvement in performance on the commonly used test data, such as Penn Treebank section 23, between 2005 and 2007 motivated SRL researchers to seek help from the grammatical relations between a word and its head within the framework of dependency grammar (Johansson and Nugues, 2007a,b; Surdeanu et al., 2008). However, the shift in feature representations, aiming to show that dependency grammar was more suitable for the feature design, was not

* The author would like to thank the reviewers for their valuable comments. The author would also like to thank Professors Steven Abney and Richmond Thomason and Mr. Terry Szymanski for their help.

convincingly supported by the results from CoNLL-2008. As a matter of fact, the researchers concluded, post CoNLL-2008, that determining the right syntactic/semantic knowledge for the SRL task still remained an open question (Surdeanu et al., 2008) and hence would be further explored on CoNLL-2009.

The preceding conclusion and the trend in feature engineering indicate that finding the right syntactic/semantic knowledge remains a challenge. At the same time, how to generalize across different syntactic configurations involving the same verb remains an open question as well.

To provide a satisfying solution to the preceding question, we first identify the syntactic configurations that make it challenging for SRL researchers to create generalizable features. These configurations involve moved or displaced arguments. Then, relying on the knowledge of argument structure of the verb and on the context dependence among the core semantic arguments defined in the theory of argument realization (Levin and Hovav, 2005), we design the base argument configuration (BAC) features that generalize across the challenging syntactic configurations. Together with two-levels of backoff features, the BAC features effectively help with the argument role labeling subtask.

2 Basic definitions: core and non-core semantic arguments

We begin with defining the major terms in this paper. Following PropBank (Palmer et al., 2005), the present work defines a semantic argument as one of the constituents that participate in the event that the predicate is involved in, including the syntactic arguments, adjuncts, modal verbs, negation adverbs, and discourse markers. The core semantic arguments correspond to the syntactic arguments, and the non-core semantic arguments refer to the adjuncts, modal verbs, and discourse markers. In the rest of the paper, any mention of *argument* refers to *semantic argument*, unless otherwise specified. And we refer to the argument in syntactic structures specifically *syntactic argument*.

3 The challenge

We motivate the current project by depicting the fact that little had been done to design features to generalize across different syntactic structures that a verb occurs in. In the past eight years, SRL researchers have increased the number of features from the initial five in Gildea and Jurafsky (2000) to over twenty in later work such as (Pradhan et al., 2004; Toutanova et al., 2005; Haghighi et al., 2005; Surdeanu et al., 2007). While a large portion of these provide rich representation of the lexical and syntactic features of the arguments, more and more features are engineered towards specific syntactic structures. Punyakanok et al. (2005) imposed seven constraints on seven types of surface syntactic structures. Having found that the previous systems had not properly handled the cases where two consecutive arguments could take the same semantic role in relative clauses and the noun phrase it modifies, Toutanova et al. (2005) created the repeated core argument label feature. Moreover, Haghighi et al. (2005) came up with the projected path feature to handle the moved subject in the control-verb structures. Interestingly, it was Gildea and Jurafsky (2000) who first started this trend of tailoring features to specific syntactic structures. In their seminal work, Gildea and Jurafsky (2000) created the *voice* feature to distinguish between the passive and active configurations involving a verb. In the following, we illustrate the base argument configuration feature that generalizes across different surface syntactic structures. Intuitively, features generalize over different structures may increase the coverage of the classifier. In the next section, we introduce the basic syntactic and lexical semantic concepts that lend background to our solution to the challenge.

4 Syntactic and lexical semantic background

In this section, we introduce the syntactic and lexical semantic knowledge that allow us to create a feature design that generalizes over the diverse syntactic variations involving a verb.

4.1 Argument structure and base argument configuration

The arguments of a verb refer to its subject/external argument and complement(s)/internal argument(s). Argument structure refers to the number of external and internal arguments that a particular predicate verb requires in a clause (Carnie, 2002, p. 166). This definition indicates that the argument structure of a verb not only specifies the number of syntactic arguments it takes but also determines their positions in the structure. Based on the number of possible arguments and their positions in the argument structure of a verb in a single clause, Quirk et al. (1985, p. 53) summarize three main types of argument structures, including the 2-element structure *Subject-V*, the 3-element structure *Subject-V-O/AdjP/PP complement*, and the 4-element structure *Subject-V-O-O/AdjP/PP complement*. It is possible that a verb has more than one argument structure. In this paper, we name the subset of argument structures of the verbs in active voice learned from Treebank as **base argument configuration**.

4.2 Variations from base argument configuration

From Treebank, our system also learns four types of structures where some argument is either moved or displaced so that they are not present at the position in the base argument configuration they are expected to be at. Argument movement and displacement cause structural variations from argument structures or base argument configurations. This is the reason why previous SRL systems created specific features to account for the variations, such as the *passive* feature to account for the arguments of a passive verb. The knowledge of these structures forms the basis of our system. The structures include the ones involving movements, displaced arguments, shared arguments, and extra arguments.

4.3 Lexical semantic background: context dependence

In addition to the preceding syntactic background, the current system also draws up the *theory of argument realization* (Levin and Hovav, 2005), the updated *Linking theory* (Levin and Hovav, 1996). The latter has been the linguistic theoretical basis since (Gildea and Jurafsky, 2000; Gildea and Palmer, 2002). Levin and Hovav (2005) extended the original linking theory in many aspects. We utilize the context dependence among the semantic roles of core semantic arguments in the updated theory by enforcing the context dependence constraint as follows.

Context dependence constraint: The semantic roles of the core semantic arguments of a given verb that are dependent on each other are realized through a specific syntactic configuration of the core semantic arguments of the verb, within which the positions of the arguments assume fixed positions relevant to the verb and other arguments.

5 Feature design

With the foregoing background knowledge, we realize that both base argument configuration and context dependence concern the core semantic arguments and that the moved and displaced arguments are also core semantic arguments. Therefore, we decide to create different features for core and non-core arguments. For the core arguments, we propose the base argument configuration features and two levels of backoff features. For the non-core arguments, we extract a generic set of commonly used features which happened to be the second level of backoff features.

5.1 Base argument configuration features

The base argument configuration (BAC) feature is a sequential listing of the core semantic arguments of a predicate verb corresponding with their positions in the verb's base argument configuration/argument structure, with the verb inserted at the correct position in the listing. The BAC feature not only captures the base argument configuration of the verb but also imposes the context dependence constraint by ensuring all realized arguments to be present at their expected positions.

Level-II feature:

phrase type - the syntactic category of the constituent **path** - the path from the constituent to the predicate
voice - the voice of the clause where the constituent is in **position** - position with respect to the predicate
head word POS - POS of the head word of the constituent **first word/POS** - the first word/POS of the constituent
last word/POS - the last word/POS of the constituent **predicate** - the predicate verb itself
parent head word/POS - the head word of the constituent's parent node and its POS
right sibling phrase type/head/POS - the syntactic category of the constituent's right sibling node, its head word, the head's POS
left sibling phrase type/head/POS - the syntactic category of the constituent's left sibling node, its POS and head word

Figure 1: Level-II features

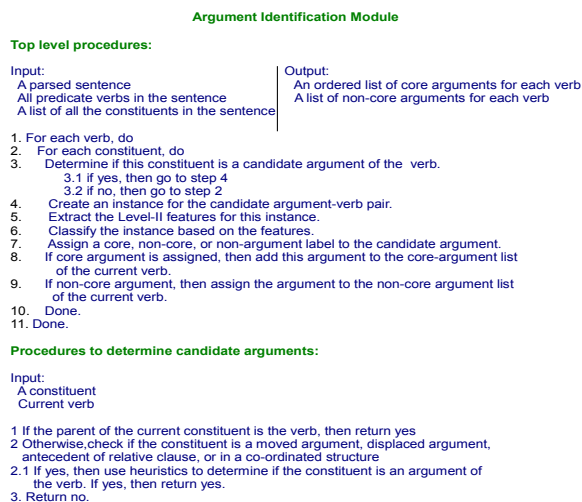


Figure 2: Argument Identification Module

5.2 Level-I features: handling unrealized core arguments

If any of the argument of the verb is unrealized, the Level-I representing the current arguments and their positions in the base argument configuration is extracted.

5.3 Level-II features: handling unknown verbs

Both BAC and Level-I features are centered on the predicate verb. There are cases where the predicate during testing is unknown. In these cases, a set of generic features commonly used in literature are used (Pradhan et al., 2004; Toutanova et al., 2005). This set consists the features shown in Figure 1. The Level-II features are also extracted for non-core arguments, according to their expected positions in the verb's argument structure.

6 System architecture

The current system proceeds with an argument identification phase and an argument classification phase.

6.1 Argument identification module

Unlike existing SRL systems that explicitly or implicitly identify the constituents as arguments or non-arguments, the current argument identifier performs a three-way identification. That is, a candidate is identified as either a core argument, non-core argument, or non argument. The purpose of identifying the arguments as core and non-core arguments is to be able to assign different sets of features to them. The argument identifier is summarized in Figure 2.

Argument Classification Module

Main Procedures:

Input:

- A sentence
 - lists of core arguments, one for each verb, ordered by their current positions in the sentence
 - lists of non-core arguments, one for each verb
1. For each verb, do
 2. Normalize the order of the core arguments in its core-argument list.
 3. For each verb in the current sentence, do:
 1. For each argument of the verb, do:
 2. Create an instance for the argument-verb pair.
 6. If the current argument is a **core argument**, then do
 - 6.1. Extract the BAC feature for the current instance.
 - 6.2. If there is unrealized argument(s), extract the Level-I feature
 - 6.3. If the verb is new verb, extract the Level-II features.
 - 6.4. Go to step 8.
 7. If the current argument is a **non-core argument**, then extract the Level-II features for the current instance.
 8. Classify the current instance using the extracted feature(s) and assign the semantic role to the argument.
 9. Done.
 10. Done.

Procedures to normalize argument positions:

Input: an ordered list of core arguments of a predicate verb

Output: an ordered list with the positions normalized

1. If the current clause is one of the nine types of syntactic configuration involving moved arguments, then
 - 1.1 Identify its originating position using the knowledge about the syntactic configuration.
 - 1.2 Move the argument to the originating position by rearranging its position in the list.

Figure 3: Argument Classification Module

6.2 Argument classification module

The argument identification module returns a list of core arguments and a list of non-core arguments for each verb. For each non-core argument, the classification module extracts the Level-II features, classifies them, and assigns them the corresponding semantic role label. For the core arguments, the critical step is to normalize the arguments' positions. If any argument appears in one of the structures involving moved or displaced argument, the argument will be returned to its originating position based on the built-in knowledge of the moved/placed argument positions in these structures.

7 Experiments, results, and discussion

We trained a three-way argument identifier using logistic regression with L2 regularization. We also trained KNN and L2 logistic regression argument classifiers with three different settings of features. WSJ section 24 was our development data. We performed three sets of experiments with WSJ section 23. We report the results from the three sets of experiments with WSJ section 23 in the following.

7.1 Experiments with argument classification only

This set of experiments assumed that core and non-core arguments were given and only performed the argument role assignment task. The goal of this set of experiments is to test our system and feature designs for the role assignment task. Table 1 lists the argument classification results from this set of experiments. The classifiers in Experiment #1 were trained/tested using the BAC features for each argument. Level-I features were added to the classifiers in Experiment #2. And Level-II features were added to Experiment #3. The Baseline classifier was trained/tested using only the Level-II features. The re-ranking system was an re-implementation of the joint-inferencing model in Toutanova et al. (2005) by Surdeanu et al. (2007), one of the top models in the

CoNLL-2005 SRL shared task. Surdeanu et al. (2007) also built system using combined strategies, which remains the state-of-the-system using constituency grammar representations. Since the KNN classifier did a slightly better job than the LR model, all scores about our system refer to those of the KNN models in the following.

Table 1: Classification results from baseline model, logistic regression (LR) and KNN models, and two state-of-the-art systems

Experiments	Baseline	1	2	3	Re-ranking	Combined strategies
Classifier	LR	LR KNN	LR KNN	LR KNN	Joint Inference	Combined strategies
Features for core arguments	Level-II features only	BAC features	BAC + Level-I + features	BAC + Level-I + Level-II features	Level-I+ others	Level-I+ others
Ftrs for non-core	Level-II	Level-II	Level-II	Level-II		
Precision	82.69	91.83 92.42	91.54 92.10	88.73 89.24	88.08	99.12
Recall	83.15	84.25 84.79	84.94 85.45	89.06 89.57	82.84	85.22
F-measure	82.92	87.88 88.44	88.12 88.65	88.89 89.41	85.38	91.64

We could make the following conclusions from our experiments.

The BAC features alone were effective because Experiment #1 gets higher precision, recall, F-measures than both the baseline and the re-ranking model. The Level-I features helped Experiment #2 get higher recall than the combined strategies, while increasing the overall F-measure. The Level-II features expectedly helped Experiment #3 with recall and F-measure, showing the increased coverage of the Level-II features. One more note we would make for this set of experiments is that the argument classifier per se is quite effective and consistent as will be shown in the next two sets of experiments.

7.2 Identification and classification with gold parses

In this set of experiments, we started with an LR argument identification module that identified the core and non-core arguments from the Treebank gold parses. We then ran the top performing KNN argument classifier from Table 1 on the automatically identified core and non-core argument lists. Table 2 lists the results.

Table 2: Experimental results with the gold parses

Experiment	Precision	Recall	F-measure
Baseline system			
Arg. Identification arg or non-argument	92.16	89.47	90.80
Arg. Id. & Classification	73.22	71.90	72.55
Current system			
Arg. Identification core, non-core, non args.	90.31	86.42	88.32
Arg. Id. & Classification	81.16	78.23	79.67
% of classification correctness from identification	89.87	90.52	90.20

We could make the following observations from the results. Although the identification and classification results are higher than the baseline system, the performance of the argument classifier is greatly affected by that of the argument identifier. However, judging by the percentage classification correctness of 89.87, 90.52, and 90.20 in precision, recall, and F-measure from the identification results, the argument classifier performs consistently because the percentage correctness in all three measures matches that of the top performing argument classifier in Table 1.

7.3 Identification and classification with automatic parses

In this set of experiments, we fed the argument identifier with automatically parsed input using Charnaik parser McClosky et al. (2006). We then ran the top performing KNN argument classifier from Table 1 on the identified core and non-core arguments. The results are shown in Table 3.

Table 3: Experimental results with the automatically parsed input

Experiment	Precision	Recall	F-measure
Baseline system			
Arg. Identification arg or non-argument	82.51	80.25	81.36
Arg. Id. & Classification	66.13	65.04	65.58
Current system			
Arg. Identification core, non-core, non args.	80.88	77.36	79.08
Arg. Id. & Classification	72.20	69.57	70.86
% of classification correctness from identification	89.26	89.93	89.60

We observed similar trends as in the preceding set of experiments. The argument classifier is affected greatly by the identifier. The fact that the final system F-measure is several percent lower than the state-of-the-art system makes the argument identification results unacceptable. Despite the performance of the argument identifier, Table 3 shows that the argument classifier remains consistent because the percentage correctness of 89.26 89.93 and 89.60 in precision, recall, and F-measure from the identification results again matches that of the best argument classifier from Table 1.

8 Related work

As far as we know, the current system is the first one that specifically explores a feature design that generalizes across different syntactic configurations that a predicate occurs in. It has been a long history that SRL systems have tried to use the dependence among semantic arguments, such as (Gildea and Palmer, 2002; Xue and Palmer, 2004; Toutanova et al., 2005). However, these systems represent the dependency by simply listing all the arguments of a verb without leaving out the non-core arguments because context dependence only exists among the core arguments. The current system distinguish between the core and non-core arguments, which makes the BAC features high effectively, as shown in the experiments. At the same time, these systems do not identify moved or displaced arguments. Representing context dependence by simply listing the moved or displaced arguments without resolving their original positions may not precisely represent the dependency in the argument structure of the verb.

Vickrey and Koller (2008) applied sentence simplification to transform each clause in the sentence to a simple sentence before assign semantic roles to the constituents. They had to write about 150 rules to transform the sentences. We rely on the knowledge of the structures involving moved and displaced argument to reconstruct the original positions of the arguments. But, our set of heuristics is far fewer than 150. In addition, Vickrey and Koller (2008) did not perform argument identification while we do.

9 Conclusion and future work

In this paper, we showed that the base argument configuration features, based on the knowledge of argument structure, on the constraint imposed by context dependence defined in the theory of argument realization, and on the knowledge of moved and displaced core arguments, and the two levels of features are robust in terms of solving the argument classification task. However, we have yet to improve the argument identifier in order to achieve the overall system performance.

References

- Carnie, Andrew. 2002. *Syntax, A Generative Approach*, Blackwell Publishing.
- Carreras, Xavier and Lluís Marquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL)*, Ann Arbor, Michigan, 152–164.

- Gildea, Daniel and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Conference of the Association for Computational Linguistics (ACL-00)*, Hong Kong, 512–520.
- Gildea, Daniel and Martha Palmer. 2002. The necessity of syntactic parsing for predicate argument recognition. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*, Philadelphia, PA, 239–246.
- Haghighi, Aria, Kristina Toutanova, and Christopher Manning. 2005. A joint model for semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, Ann Arbor, Michigan, 173–176.
- Johansson, Richard and Pierre Nugues. 2007a. Extended constituent-to-dependency conversion for english. In *Proceedings of NODALIDA 2007*.
- Johansson, Richard and Pierre Nugues. 2007b. Lth: Semantic structure extraction using nonprojective dependency trees. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, 227–230.
- Levin, Beth and Malka Rappaport Hovav. 1996. From lexical semantics to argument realization, *Manuscript*, 1, 1–8.
- Levin, Beth and Malka Rappaport Hovav. 2005. *Argument Realization*, Cambridge.
- McClosky, David, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the North American Conference on Computational Linguistics*.
- Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31, 71–106.
- Pradhan, Sameer, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Association for Computational Linguistics annual meeting (HLT/NAACL-2004)*.
- Punyakanok, Vasin, Dan Roth, and Wen tau Yih. 2005. The necessity of syntactic parsing for semantic role labeling. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*.
- Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*, Longman.
- Surdeanu, Mihai, Richard Johansson, Lluís Marquez, Adam Meyers, and Joakim Nivre. 2008. CoNLL-2008 shared task description, online.
- Surdeanu, Mihai, Luis Marquez, Xavier Carreras, and Pere R. Comas. 2007. Combination strategies for semantic role labeling. *Journal of Artificial Intelligence Research*, 29, 105–151.
- Toutanova, Kristina, Aria Haghighi, and Christopher D. Manning. 2005. Joint learning improves semantic role labeling. In *Proceedings of ACL 2005*, Ann Arbor, MI.
- Vickrey, David and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*.
- Xue, Nianwen and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of EMNLP*, Barcelona, Spain.