Semantic Role Labeling using Linear-Chain CRF

Melanie Tosik

University of Potsdam, Department Linguistics

Seminar: Advanced Language Modeling (Dr. Thomas Hanneforth)

September 22, 2015

Abstract

The aim of this paper is to present a simplified take on applying linear-chain conditional random fields (CRF) to semantic role labeling (SRL), with a focus on German. The dataset is adapted from the semantic parsing track of the CoNLL-2009 shared task on syntactic and semantic dependencies in multiple languages. By treating SRL as a sequence labeling task, the framework architecture becomes very simple. Building on a set of hand-crafted features, a linear-chain CRF model is trained which jointly performs argument identification and classification in a single step. The best results on the sequence tagging task are obtained by the model which integrates basic argument and predicate features, as well as a binary feature indicating if a given argument is a syntactic child of the predicate in the dependency tree. We found that for our system, employing more distinct features on syntactic dependents of the predicate impaired model performance.

1 Introduction

In natural language processing (NLP), SRL (sometimes also called case role analysis, thematic analysis, or shallow semantic parsing) refers to the task of identifying the semantic arguments of each predicate (typically the verb) in the sentence, and classifying them into their predicate-specific semantic roles. Dating back to Fillmore (1968), semantic roles originated in the linguistic notion of case. Common semantic role labels include *Agent* (actor of action), *Patient* (entity affected by the action), *Instrument* (tool used to perform action), *Beneficiary* (entity for whom action is performed), *Source* (origin of the effected entity), or *Destination* (destination of the affected entity). For example:

$$[\text{ John }]_{\text{AGENT}} \text{ hit } [\text{ Mary }]_{\text{PATIENT}} [\text{ with a stick }]_{\text{INSTRUMENT}} .$$

To date, SRL has been successfully applied to a variety of NLP tasks. Most commonly, it is used in questing answering (QA) systems, where semantic arguments can frequently answer the questions of *Who?*, *What?*, *How?* etc., and machine translation (MT), where semantic roles are usually expressed using language-specific syntactical structures. Because of this correlation between syntax and semantics, syntactic positions of predicate arguments tend to be good indicators of the semantic role they play in the sentence. For example, while subjects are often agents, direct objects are likely to be patients, and objects of *with*-prepositional phrases (PPs) are probably instruments (just like in the example above).

However, SRL is not a trivial problem. In order to build a complete SRL system, it is necessary to determine the correct parse tree for each sentence, as well as the correct word senses and the corresponding semantic roles. Word sense disambiguation is a crucial prerequisite to argument classification because semantically ambiguous words may require different numbers and realizations of semantic roles for each possible word sense. For example, the English verb *walk* can take one to three, and possibly even more semantic arguments, depending on the context:

(1) John walks home.

(2) John walks the dog.

(3) John walks the dog to the vet.

Typically, statistical methods are used to automatically acquire and apply the complex knowledge that is needed for effective and efficient SRL systems. To this end, many of the standard machine learning techniques can be employed with varying success rates. For example, in the CoNLL-2005 Shared Task[1] on PropBank SRL (Kingsbury and Palmer, 2002), 19 teams participated with a wide range of learning approaches, including maximum entropy (MaxEnt), support vector machine (SVM), SNoW (an ensemble of enhanced perceptrons), decision trees, AdaBoost (an ensemble of decision trees), nearest neighbor, tree conditional random field (CRF), as well as different combinations of these approaches.

2 Conditional Random Fields (CRF)

Conditional random fields (CRF) is a state-of-the-art sequence labeling framework introduced by Lafferty et al. (2001). CRF is an undirected, graphical model, which is trained to maximize a conditional probability distribution over a given set of features.

The most common graphical structure used with CRF is linear-chain, a special case of general CRF restricted in that every output label y_i only depends on the h labels that are directly preceding it (in practice, h is usually set to 1). Assume $Y = (y_1, ..., y_T)$ denotes a sequence of labels, and $X = (x_1, ..., x_T)$ denotes the corresponding observations sequence. The sequence of labels is the concept we wish to predict, e.g. named-entities, part-of-speech (POS) tags, or semantic role labels. The observations are the strings in the input sequence. Given a linear-chain CRF, the conditional probability $p(Y|X)$ is then computed as

$$p(Y|X) = \frac{1}{Z_X} \prod_{t=1}^{T} exp \left\{ \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t) \right\},$$

where Z_X is a normalizing constant such that all the terms normalize to one, f_k is a feature function, and λ_k is a feature weight. CRF offers an advantage over generative approaches such as hidden Markov models (HMMs) by relaxing the conditional independence assumption and allowing for arbitrary features in the observation.

For all our experiments we use *CRFsuite*[2], an implementation of CRF for labeling sequential data provided by Okazaki (2007). We choose an appropriate learning algorithm based on accuracy on the test set and use Limited-memory BFGS optimization (Nocedal, 1980).

[1] `http://www.cs.upc.edu/~srlconll/`

[2] `http://www.chokkan.org/software/crfsuite/`

3 Experimental setup

We start by describing our datasets in Section 3.1. Section 3.2 details the feature sets implemented in the models. Section 3.3 specifies how the models are evaluated.

3.1 Data

The dataset is adapted from the CoNLL-2009 Shared Task on syntactic and semantic dependencies in multiple languages[3]. Since only training and development data are still freely available for German, the development set is used as test set. A detailed description of the CoNLL-2009 data format can be found on the task website. In short, annotated data in dependency format is provided for statistical training, where the dependency labels have been extracted from manually annotated treebanks such as the German TIGER Treebank (Brants et al., 2002). The dependency trees have additionally been enriched with semantic labels and relations such as those captured in the PropBank and similar resources.

An overview of the data columns is given below. P-columns are automatically predicted variants of the gold-standard LEMMA, POS, FEAT, HEAD, and DEPREL columns produced by independently (or cross-)trained taggers and parsers. FEAT is a set of morphological features (separated by |) defined for a particular language. FILLPRED contains Y for lines where PRED is filled. PRED is the column for the predicate along with its specific verb sense. APRED contains the semantic roles. _ is used for unknown, unannotated, unfilled, etc. values.

Gold fields	ID FORM LEMMA POS FEAT HEAD DEPREL
Predicted fields	PLEMMA PPOS PFEAT PHEAD PDEPREL
Additional fields	FILLPRED PRED APREDs

The original CoNLL-2009 Shared Task objective was to perform and evaluate SRL using a dependency-based representation for both syntactic and semantic dependencies, for predicates of all major POS categories. Due to a limited availability of the extensive resources needed to recreate the exact task, several simplifying changes have been made to the original datasets. First, both datasets are pre-processed to only contain sentences with exactly one verb predicate. Sentences with more than one predicate rarely occur in the data, and filtering them eases the computation of the corresponding semantic role labels. Table 1 provides an overview of the number of predicates per sentence for each dataset.

Furthermore, the main focus of this work is on labeling argument candidates with a predicate-specific semantic role. Therefore, instead of automatically determining the word sense for each predicate, gold annotations for each predicate were given as input to our system.

Last, we limit the semantic role label set to A0-A9 (following the PropBank label set), corresponding to the number of possible semantic arguments for each predicate, where the A0 label is usually assigned to arguments which are understood as agents, the A1 label is assigned to the patient argument, and so on.

[3] `http://ufal.mff.cuni.cz/conll2009-st/index.html`

# Predicates	Training set	Test set
none	21738	1468
1	11562	480
2	2370	48
3	311	3
4	31	(n/a)
5	7	(n/a)
6	1	(n/a)

Table 1: Number of sentences with their number of predicates for training and test set.

3.2 Features

As indicated in Section 2, the CRF model learns based on a number of pre-defined features. In the case of SRL, the model tries to extract a semantic role for each argument candidate. Since we are dealing with a dependency representation of the data, no pruning is done to obtain a pre-defined set of syntactic constituents that are likely to be argument candidates. Instead, every input word is individually considered a potential semantic argument to the predicate.

Extracting the right set of features is crucial for successfully applying any machine learning algorithm. In order for the learning algorithm to discover truly relevant patterns in the data, we have to provide it with domain specific knowledge and, ultimately, human insight. Thus, for the SRL labeling task, an obvious set of features will at least contain the word form, lemma, POS, and morphological features for each word, as well as its dependency relation to the predicate (recall that subjects, for instance, are likely to be semantic agents). We automatically extract these features from the training data, and use them as argument baseline features.

In the next step, we identify the verb predicate for each sentence, and enhance the model by integrating the form, lemma, POS, morphological features, and the DEPREL value of the predicate as predicate features. Moreover, we define a binary feature indicating if the current word is the sentence predicate or not. In addition, we introduce a binary flag which is true if the current word is a syntactic child of the predicate, and false otherwise. We also experimented with incorporating the full set of features for each syntactic child (form, lemma, POS, morphological features, dependency relation) as predicate children features, and splitting the morphological features in FEAT into its different individual features, e.g. gender, case, number, etc.

3.3 Evaluation

We evaluate five different models based on the features above (cf. Table 2).

The first baseline model uses only the argument baseline features. For the following models, we add the predicate features, the binary *Is child?* feature, the predicate children features, as well as the field splitting of the morphological features, respectively.

However, since the majority of predicates do not take more than three semantic arguments, most words in any given sentence are not going to be assigned a semantic role (but _ instead). Therefore, if the model was to simply assign _ to every single word, the overall model accuracy would still be fairly high. To prevent the results from being distorted, we thus evaluate exact match precision, recall, and F1 score for each label individually. Labels A0 and A1 are the most frequent labels, and equally distributed over the test set (360 and 361 occurrences, respectively).

While 74 instances of label A3 are present in the test set, label A4 is found in 19 sentences. Labels A5-A9 are discarded from the evaluation because one average, they each only occur once in the test data.

4 Results

The exact match precision, recall, and F1 scores for each label in the test set are shown in Table 2. In addition, Table 3 contains a description of the features implemented in the CRF models. Note that, except for the baseline model, each model builds on the previous one(s), thus extending the feature space with every new model, not entirely replacing it.

	Test set [%]														
	A0			A1			A2			A3			A4		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
#1	56.0	58.3	57.1	42.6	19.9	27.2	20.0	0.8	1.4	0.0	0.0	0.0	0.0	0.0	0.0
#2	59.7	65.8	62.6	52.0	31.6	39.3	51.2	16.4	24.9	60.0	16.2	25.5	57.14	21.0	30.8
#3	74.9	65.6	69.9	72.0	61.2	66.2	63.6	41.8	50.5	69.4	46.0	55.3	75.0	31.6	44.4
#4	62.5	57.8	60.0	59.1	46.8	52.2	0.5	28.4	36.2	38.2	17.6	24.1	46.7	36.8	41.2
#5	70.3	66.9	68.6	63.9	54.9	59.0	55.3	31.3	40.0	44.4	21.6	29.1	50.0	31.6	38.7

Table 2: Precision (P), recall (R), and F1 scores of the CRF models for each label.

Model #1	Baseline argument features
Model #2	+ Predicate features
Model #3	+ Is child? (y/n) feature
Model #4	+ Children features
Model #5	+ Field splitting for morphological features

Table 3: Model descriptions.

As can be seen, the baseline Model #1 starts off with a decent performance on identifying A0 roles (57.1% F1 score), but then rapidly gets worse for every subsequent label. Labels A3 and A4 do not get assigned at all, resulting in 0% F1 score for those roles. The second Model #2 adds the predicate features, resulting in large performance gains across all labels. The biggest improvement is concerning role A4, with a boost in F1 score of +30.8%.

Model #3 only adds a single new feature, namely a positive binary flag for every word that has been identified as a syntactic dependent (child) of the predicate. Again, we are able to increase model performance by several points in F1 score for roles A0, A1, and A4, and more than doubling the accuracy for labels A2 and A3. This is explained by the fact that the model finally has a robust indicator of which words are very likely to be semantic arguments in the first place: since the data is represented in dependency tree format, the verb predicate is generally the syntactic root of the sentence; thus, any syntactic children are directly the semantic arguments of the predicate. Model #3 gives the overall best results across all models and role labels.

As has been suggested in related work (see, for example, Björkelund et al. (2009)), we implement features for every syntactic child of the predicate in Model #4. However, these features did not seem to help the model uncover additional ties between the input sequences and their corresponding role labels. Since there is only ever a single predicate present in each sentence, adding the predicate features is a reasonable and effective way to enhance the learning algorithm. For syntactic dependents, on the other hand, flooding the model with a possibly large number of properties of individual syntactic children has the opposite effect and actually causes the model performance to drop significantly for every role label, with a loss in F1 score of up to -31.2% for label A3.

Except for A4, adding the morphological feature splitting in Model #5 brings model accuracy back up by a few points for every label. To verify the affect of the additional individual morphological features, they have been implemented in several other model architectures not mentioned here. The results did not prove effective, suggesting that the morphological features in the FEAT column are similar enough to already contribute their share if adopting the original concatenated representation.

In general, we find that the models consistently yield a higher accuracy for A0 than for every other semantic role. While this might be expected for labels A2-4, it appears significant with respect to A1. Since both labels occur equally often in the data, this could be treated as evidence that it is intrinsically harder to automatically infer the semantic patient of a sentence than it is to identify an agent. In addition, the results also confirm what has already been stated in many recent publications using linear-chain CRF architectures: namely that the system's recall performance is predominantly lower than precision accuracies. In this case, this is increasingly observed for labels A2-A4, but could be explained by the lack of a sufficient number of training examples in the training data.

5 Conclusion and future work

Semantic role labeling (SRL) remains a challenging task for researchers in natural language processing (NLP). In this paper, we presented a simple method of performing and evaluating SRL by treating it as a straightforward sequence labeling task. The extraction task is solved by integrating a number of pre-defined features into the linear-chain conditional random fields (CRF) framework introduced by Lafferty et al. (2001).

We built a SRL dataset for German based on the training and development data released in the context of the semantic parsing track of the CoNLL-2009 Shared Task. We modified the data by filtering out all sentences that did not comprise a single verb predicate only, and keeping gold predicate senses instead of automatically performing the word sense disambiguation.

We found that in our case, we obtained the best results on the extraction task by employing a cascaded model that incorporates semantic and syntactic information for every argument word, as well the sentence predicate. In addition, a binary feature for syntactic dependents of the predicate is used.

From here, there are many directions future work might take. The current system could bit by bit be extended to eventually meet all the official requirements posed by the CoNLL-2009 SRL Shared Task. Furthermore, it could be worthwhile to compare the performance of the linear-chain CRF architecture to a tree-structured CRF model, which could operate on full syntactic analyses rather than a dependency-based language representation, and thus learn to assign semantic roles to complete syntactic constituents rather than individual words.

References

Björkelund, A., L. Hafdell, and P. Nugues (2009). Multilingual Semantic Role Labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, Stroudsburg, PA, USA, pp. 43–48. Association for Computational Linguistics.

Brants, S., S. Dipper, S. Hansen, W. Lezius, and G. Smith (2002). The TIGER Treebank.

Fillmore, C. J. (1968). The Case forCase. In E. W. Bach and R. T. Harms (Eds.), *Universals in Linguistic Theory*, pp. 1–88. New York: Holt, Rinehart & Winston.

Kingsbury, P. and M. Palmer (2002). From treebank to propbank. In *In Language Resources and Evaluation*.

Lafferty, J. D., A. McCallum, and F. C. N. Pereira (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, San Francisco, CA, USA, pp. 282–289. Morgan Kaufmann Publishers Inc.

Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of computation 35*(151), 773–782.

Okazaki, N. (2007). CRFsuite: a fast implementation of Conditional Random Fields (CRFs).