A Lightweight Semantic Chunking Model Based On Tagging

Kadri Hacioglu
Center for Spoken Language Research,
University of Colorado, Boulder
hacioglu@cslr.colorado.edu

Abstract

In this paper, a framework for the development of a fast, accurate, and highly portable semantic chunker is introduced. The framework is based on a non-overlapping, shallow tree-structured language. The derivation of the tree is considered as a sequence of tagging actions in a predefined linguistic context, and a novel semantic chunker is accordingly developed. It groups the phrase chunks into the arguments of a given predicate in a bottom-up fashion. This is quite different from current approaches to semantic parsing or chunking that depend on full statistical syntactic parsers that require tree bank style annotation. We compare it with a recently proposed word-by-word semantic chunker and present results that show that the phrase-by-phrase approach performs better than its word-by-word counterpart.

1 Introduction

Semantic representation, and, obviously, its extraction from an input text, are very important for several natural language processing tasks; namely, information extraction, question answering, summarization, machine translation and dialog management. For example, in question answering systems, semantic representations can be used to understand the user's question, expand the query, find relevant documents and present a summary of multiple documents as the answer.

Semantic representations are often defined as a collection of frames with a number of slots for each frame to represent the task structure and domain objects. This frame-based semantic representation has been successfully used in many limited-domain tasks. For fully used in many limited-domain tasks. For example, in a spoken dialog system designed for travel planning one might have an ***Air*** frame with slots ***Origin***, ***Destination***, ***Depart_date***, ***Airline*** etc. The drawback of this domain specific representation is the high cost to achieve adequate coverage in a new domain. A new set of frames and slots are needed when the task is extended or changed. Authoring the patterns that instantiate those frames is time consuming and expensive.

Domain independent semantic representations can overcome the poor portability of domain specific representations. A natural candidate for this representation is the predicate-argument structure of a sentence that exists in most languages. In this structure, a word is specified as a predicate and a number of word groups are considered as arguments accompanying the predicate. Those arguments are assigned different semantic categories depending on the roles that they play with respect to the predicate. Researchers have used several different sets of argument labels. One possibility are the non-mnemonic labels used in the PropBank corpus (Kingsbury and Palmer, 2002): ARG0, ARG1, …, ARGM-LOC, etc. An alternative set are *thematic roles* similar to those proposed in (Gildea and Jurafsky, 2002): AGENT, ACTOR, BENEFICIARY, CAUSE, etc.

Shallow semantic parsing with the goal of creating a domain independent meaning representation based on predicate/argument structure was first explored in detail by (Gildea and Jurafsky, 2002). Since then several variants of the basic approach have been introduced using different features and different classifiers based on various machine-learning methods (Gildea and Palmer, 2002;.Gildea and Hockenmaier, 2003; Surdeanu et. al., 2003; Chen and Rambow, 2003; Fleischman and Hovy, 2003; Hacioglu and Ward, 2003; Thompson et. al., 2003 ; Pradhan et. al., 2003). Large semantically annotated databases, like FrameNet (Baker et.al, 1998) and PropBank (Kingsbury and Palmer, 2002**)** have been used to train and test the classifiers. Most of these approaches can be divided into two broad classes: Constituent-by-Constituent (C-by-C) or Word-by-Word (W-by-W) classifiers. In C-by-C classification, the syntactic tree

This work is supported by the ARDA AQUAINT program via contract OCG4423B and by the NSF via grant ISS-9978025

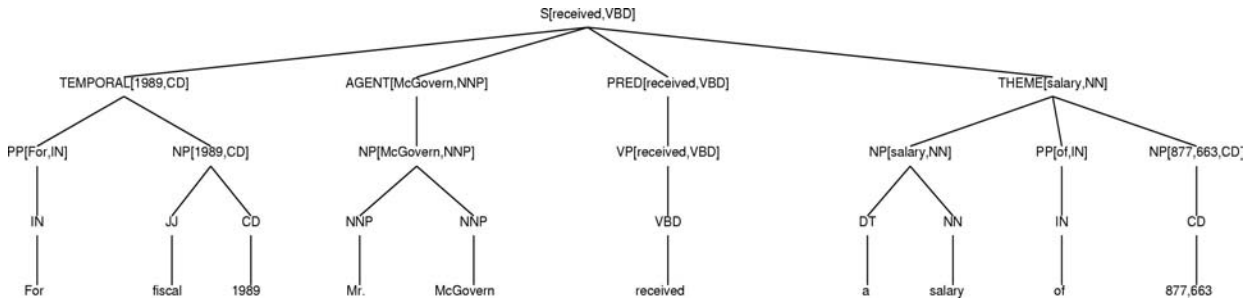

Figure 1. Proposed non-overlapping, shallow lexicalized syntactic/semantic tree structure

representation of a sentence is linearized into a sequence of its syntactic constituents (non-terminals). Then each constituent is classified into one of several arguments or semantic roles using a number of features derived from its respective context. In the W-by-W method (Hacioglu and Ward, 2003) the problem is formulated as a chunking task and the features are derived for each word (assuming part of speech tags and syntactic phrase chunks are available), and the word is classified into one of the semantic labels using an IOB2 representation. Among those methods, only the W-by-W method considered semantic classification with features created in a bottom-up manner. The motivations for bottom-up analysis are

- Full syntactic parsing is computationally expensive
- Taggers and chunkers are fast
- Not all languages have full syntactic parsers
- The annotation effort required for a full syntactic parser is larger than that required for taggers and chunkers.

In this paper, we propose a non-overlapping shallow tree structure, at lexical, syntactic and semantic levels to represent the language. The goal is to improve the portability of semantic processing to other applications, domains and languages. The new structure is complex enough to capture crucial (non-exclusive) semantic knowledge for intended applications and simple enough to allow flat, easier and fast annotation. The human effort required for flat labeling is significantly less than that required for creating tree bank style labels. We present a particular derivation of the structure yielding a lightweight machine learned semantic chunker.

2 Representation of Language

We assume a flat, non-overlapping (or chunked) representation of language at the lexical, syntactic and semantic levels. In this representation a sentence is a sequence of base phrases at a syntactic level. A base phrase is a phrase that does not dominate another phrase. At a semantic level, the chosen predicate has a number of arguments attached to it. The arguments are filled by a sequence of base phrases that span sequences of words tagged with their part of speech. We propose to organize this flat structure in a lexicalized tree as illustrated in Fig 1. The root is the standard non-terminal *S* lexicalized with the predicate. One level below, arguments attached to the predicate are organized in a flat structure and lexicalized with headwords. The next level is organized in terms of the syntactic chunks spanned by each argument. The lower levels consist of the part of speech tags and the words. The lower level can also be extended to include flat morphological representations of words to deal with morphologically rich languages like Arabic, Korean and Turkish. One can introduce a relatively deeper structure using a small set of rules at the phrasal level under each semantic non-terminal. For example, the application of simple rules in order on THEME's chunks, such as (1) combine flat **PP NP** into a right branching **PP** and then (2) combine flat **NP** with **PP** into a recursive **NP**, will result in a relatively deeper tree. Although the main focus of the paper is on the structure presented in Figure 1, we note that a deeper structure obtained by using a **small** number of simple hand-crafted rules on syntactic chunks (applied in a bottom-up manner) is worthy of further research.

3 Model for Tree Decomposition

The tree structure introduced in the preceding section can be generated as a unique sequence of derivation actions in many different ways. We propose a model that decomposes the tree into a sequence of tagging actions at the word, phrase and argument levels. In this model the procedure is a bottom up derivation of the tree that is accomplished in several steps. Each step consists of a number of actions. The first step is a sequence of actions to tag the words with their Part-Of-Speech (POS). Then the words are tagged as inside a phrase (I), outside a phrase (O) or beginning of a phrase (B) (Ramhsaw and Marcus, 1995). For example, in Figure 1, the word *For* is tagged as B-PP, *fiscal* is tagged as B-NP, *1989* is tagged as I-NP, etc. This step is followed

by a sequence of join actions. A sequence that starts with a B-tag and continues with zero or more I-tags of the same type is joined into a single tag that represents the type of the phrase (e.g. NP, PP etc.). The next step tags phrases as inside an argument, outside an argument or beginning of an argument. Finally, we join IOB argument tags as we did for base phrases.

4 Parsing Strategy

The parse strategy based on the tagging actions consists of three components that are sequentially applied to the input text for a chosen predicate to determine its arguments. These components are POS, base phrase and semantic taggers/chunkers. In the following, each component will be described along the dimensions of its (i) input, (ii) decision context, (ii) features, (iv) classifier, and (v) output.

In the first stage, the input is the sequence of words that are processed from left-to-right. The context is defined to be a fixed-size window centered around the word in focus. The features are derived from a set of word specific features and previous tag decisions that appear in the context. A Support Vector Machine (SVM) (Vapnik, 1995) as a multi-class classifier is used to label words with their POS tags[1]. In the second stage, the input is the sequence of word/tag pairs. Context is defined in the same way as in the first stage. The features are the word/tag pairs and previous phrase IOB tags that appear in the context. An SVM classifier is used to classify the base phrase IOB label. This is very similar to the set up in (Kudo and Matsumato, 2000). In the last stage (the major contribution of the paper) we group the input, context, features and decisions as shown below.

PP	For	IN	B-PP	B-TEMPORAL
NP	1989	CD	I-NP	I-TEMPORAL
NP	McGovern	NNP	I-NP	??
VP	received	VBD	B-VP	
NP	salary	NN	I-NP	current decision
PP	of	IN	B-PP	
NP	877,663	CD	B-NP	context

The input is the base-phrase labels and headwords along with their part of speech tags and positions in the base phrase. The context is –2/+2 window centered at the base phrase in question. An SVM classifies the base phrase into semantic role tags in an IOB representation using a context including the two previous semantic tag decisions. It is possible to enrich the set of features by increasing the context window, adding new sentence level and predicate dependent features, and introducing alternate organizations of the input. An alternative to our approach is the W-by-W approach proposed in (Hacioglu and Ward, 2003). We show it below:

For	IN	B-PP	B-TEMPORAL
fiscal	JJ	B-NP	I-TEMPORAL
1989	CD	I-NP	??
Mr.	NNP	B-NP	
McGovern	NNP	I-NP	
received	VBD	B-VP	current decision
a	DT	B-NP	context
salary	` NN	I-NP	
of	PP	B-PP	
877,663	CD	B-NP	

Here the labeling is carried out in a word-by-word basis. We note that the Phrase-by-Phrase (P-by-P) tagging classifies larger units, ignores some of the words (modifiers), uses effectively a wider linguistic context for a given window size and performs tagging in a smaller number of steps.

5 Experiments

All experiments were carried out using sections 15-18 of the PropBank data holding out Section-00 and Section-23 for development and test, respectively. We used chunklink[2] to flatten syntactic trees. Then using the predicate argument annotation we obtained a new corpus of the tree structure introduced in Section 2.

All SVM classifiers, for POS tagging, syntactic phrase chunking and semantic argument labeling, were realized using the TinySVM[3] with the polynomial kernel of degree 2 and the general purpose SVM based chunker YamCha[4]. The results were evaluated using precision and recall numbers along with the F metric. Table 1 compares W-by-W and P-by-P approaches. The base features described in Section 4 along with two additional predicate specific features were used; the lemma of the predicate and a binary feature that indicates the word is before or after the predicate.

Table 1. Performance comparisons

Method	Precision	Recall	F_1
W-by-W	58% **(60%)**	49% **(52%)**	53% **(56%)**
P-by-P	63% **(66%)**	56% **(59%)**	59% **(62%)**

In these experiments the accuracy of the POS tagger was 95.5% and the F-metric of the phrase chunker was 94.5%. The figures in parantheses are for gold standard

[1] Although not limited to, SVMs are selected because of their ability to manage a large number of overlapping features with a good generalization performance.

[2] http://ilk.uvt.nl/~sabine/chunklink

[3] http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM

[4] http://cl.aist-nara.ac.jp/~taku-ku/software/yamcha

(i.e. POS and phrase features are derived from hand-annotated trees). The others show the performance of the sequential bottom-up tagging scheme that we have described in section 4. We experimented with a reduced set of PropBank arguments. The set contains the most frequent 19 arguments in the corpus.

It is interesting to note that there is a huge drop in performance for "chunked" semantic analysis as compared with the performances at mid 90s for the syntactic and lexical analyses. This clearly shows that the extraction of even "chunked" semantics of a text is a very difficult task and still a lot remains to be done to bridge the gap. This is partly due to the difficulty of having consistent semantic annotations, partly due to the missing information/features for word senses and usages, partly due to the absence of world knowledge and partly due to the relatively small size of the training set. Our other experiments clearly show that with more training data and additional features it is possible to improve the performance by 10-15% absolute (Hacioglu et. al., 2004). The feature engineering for semantic chunking is open-ended and the discussion of it is beyond the scope of the short paper. Here, we have illustrated that the P-by-P approach is a promising alternative to the recently proposed W-by-W approach (Hacioglu and Ward, 2003).

6 Conclusions

We have developed a novel phrase-by-phrase semantic chunker based on a non-overlapping (or chunked) shallow language structure at lexical, syntactic and semantic levels. We have implemented a baseline system and compared it to a recently proposed word-by-word system. We have shown better performance with the phrase-by-phrase approach. It has been also pointed out that the new method has several advantages; it classifies larger units, uses wider context, runs faster. Prior work has not considered this bottom-up strategy for semantic chunking, which we claim yields a lightweight, fast, and robust chunker at moderately high performance. Although we have flattened the trees in the PropBank corpus for our experiments, the proposed language structure supports flat annotation from scratch, which we believe is useful for porting the method to other domains and languages. While our initial results have been encouraging, this work must be extended and enhanced to produce the quality of semantic parse produced by systems using a full syntactic parse.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe 1998. The Berkley FrameNet Project. Proceedings of Coling-ACL, pp. 86-90.

John Chen and Owen Rambow. 2003. Use of Deep Linguistic Features for the Recognition and Labeling of Semantic Arguments. In *Proceedings of EMNLP-2003*, Sapporo, Japan.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28:3, pages 245-288.

Daniel Gildea and Martha Palmer. 2002. The necessity of syntactic parsing for predicate argument recognition. In *Proceedings of ACL'02.*

Daniel Gildea and Julia Hockenmaier. 2003. Identifying Semantic Roles Using Combinatory Categorical Grammar. In *Proceedings of EMNL'03, Japan.*

Micheal Fleischman and Eduard Hovy. 2003. A Maximum Entropy Approach to FrameNet Tagging. *Proceedings of HLT/NAACL-03.*

Kadri Hacioglu and Wayne Ward. 2003. Target word Detection and semantic role chunking using support vector machines. *Proceedings of HLT/NAACL-03.*

Kadri Hacioglu, Sameer Pradhan, Wayne Ward, James H. Martin and Daniel Jurafsky. 2004. Semantic Role Labeling by Tagging Syntactic Chunks. *CONLL-2004 Shared Task.*

Paul Kingsbury, Martha Palmer, 2002. From TreeBank to PropBank. *Conference on Language Resources and Evaluation LREC-2002.*

Taku Kudo, Yuji Matsumato. 2000. Use of support vector learning for chunk identification. *Proc. of the 4th Conference on Very Large Corpora,* pp. 142-144.

Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James H. Martin, Dan Jurafsky.2003. Semantic Role Parsing: Adding Semantic Structure to Unstructured Text. In *Proceedings of ICDM 2003*, Melbourne, Florida.

Lance E. Ramhsaw and Mitchell P. Marcus. 1995. Text Chunking Using Transformation Based Learning. *Proceedings of the 3rd ACL Workshop on Very Large Corpora*, pages 82-94.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using Predicate-Argument Structure for Information Extraction. *Proceedings of the 41th Annual Conference on the Association for Computational Linguistics (ACL-03).*

Cynthia A. Thompson, Roger Levy, and Christopher D. Manning. 2003. A Generative Model for Semantic Role Labeling. *Proc. of the European Conference on Machine Learning (ECML-03)*.

Vladamir Vapnik 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, USA.