Target Word Detection and Semantic Role Chunking using Support Vector Machines

Kadri Hacioglu, Wayne Ward
Center for Spoken Language Research
University of Colorado at Boulder
{hacioglu,whw}@cslr.colorado.edu

Abstract

In this paper, the automatic labeling of semantic roles in a sentence is considered as a chunking task. We define a semantic chunk as the sequence of words that fills a semantic role defined in a semantic frame. It is straightforward to convert chunking into a tagging task using one of several IOB representations. Using this representation each word is tagged with I, which means that the word is inside a chunk, or with O, which means that the word is outside a chunk, or B, which means that the word is the beginning of a chunk. Tagging can also be seen as a multi-class classification problem. After recasting the multi-class problem as a number of binary-class problems, we use support vector machines to implement the binary classifiers. We explore two semantic chunking tasks. In the first task we simultaneously detect the target word and segments of semantic roles. In the second task, in addition, we label the semantic segments with their respective semantic role types. For both tasks, we present encouraging results of experiments carried out using the annotated FrameNet database.

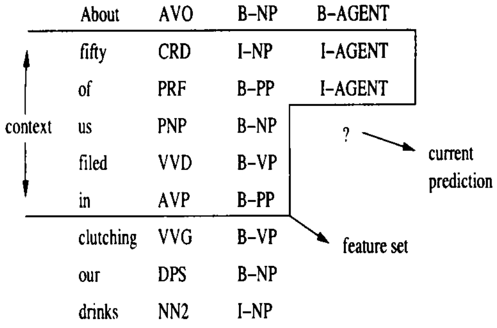
1 Introduction

Semantic representation plays a very important role in natural language interfaces between humans and computers. In simple information query and transaction tasks it is used to understand the user's input and manage the interaction. For more complex tasks, e.g. a question/answering task, the semantic representation is used to understand the question, to expand the query, to find relevant documents that match the question and to present a summary of multiple documents as the answer. For limited domain applications, a relatively simple representation of meaning, known as frame based representation, has been succesfully used. In such cases, one can define a set of frames and slots that covers the task structure and domain objects; e.g. air frame of a travel planning task with slots from_city, to_city, depart_time, depart_date, airline etc. However, for another task one has to hand-craft a new set of frames and slots. To overcome the low coverage and poor portability of this approach, we need to define semantic units at a more abstract level and avoid time consuming hand encoding of semantic knowledge. The steps in creating such a model are

- Decide on the type of semantic knowledge required
- Develop a representation to encode it
- Prepare annotated data
- Design a method to acquire that knowledge by a machine

We are interested in semantic knowledge that allows us capture, represent and understand the "Who did what to whom, where, when, how and why" in a sentence. We believe that the semantic representation by defining abstract domain independet semantic roles like AGENT, EXPERIENCER, INSTRUMENT, TIME, LOCATION, GOAL etc. fulfills our need. Availability of two databases annotated with roles, namely the FrameNet (Baker et al., 1998) and PropBank (Kingsbury and Palmer, 2002) databases, allow us train a statistical "meaning extractor"; it can be a general purpose parser for deep syntactic and semantic analysis, or a chunker for shallow semantic analysis. In this paper, we focus on shallow analysis and propose a semantic role chunker based on support vector machines (Burges, 1998).

Our work is inspired by the work in (Kudo and Matsuamto, 2000) for chunking sentences into their syntactic constituents. We extend their work to the chunking of semantic constituents in a sentence. This approach is quite



About	AVO	B-NP	B-AGENT
fifty	CRD	I-NP	I-AGENT
of	PRF	B-PP	I-AGENT
us	PNP	B-NP	
filed	VVD	B-VP	
in	AVP	B-PP	
clutching	VVG	B-VP	
our	DPS	B-NP	
drinks	NN2	I-NP	

Figure 1: Feature extraction

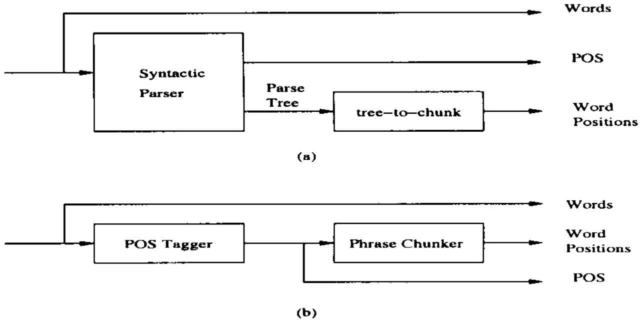


Figure 2: Two possible implementations of feature extraction

different from a related work that has been proposed in (Gildea and Jurafsky, 2002). Although we use the same FrameNet semantic representation and train our models on the same database, our method differs in the selection and creation of features, and in the implementation of the classifier.

2 Description of Chunking Method

The following example is taken from the FrameNet database:

Funny joke said [$_{AGENT}$Garvey] [$_{TARGET}$ clutching] [$_{PATIENT}$ his angel] [$_{LOCATION}$ by the shirt]

Chunks are semantic roles that have been filled by one or more words. Note that one can also consider the label TARGET as a semantic role for the sake of unified discussion. Using IBO2 representation (Sang and Veenstra, 1999), we can formulate two different chunking problems:

[$_O$Funny] [$_O$joke] [$_O$said] [$_B$Garvey] [$_{TARGET}$ clutching] [$_B$ his] [$_I$angel] [$_B$ by] [$_I$ the] [$_I$shirt]

[$_O$Funny] [$_O$joke] [$_O$said] [$_{B-AGENT}$Garvey] [$_{TARGET}$ clutching] [$_{B-PATIENT}$ his] [$_{I-PATIENT}$ angel] [$_{B-LOCATION}$ by] [$_{I-LOCATION}$ the] [$_{I-LOCATION}$ shirt]

Here each word in a sentence is labeled with a tag. I means that the word is inside a semantic role, O means that the word is outside a semantic role, and B means that the word is the beginning of a semantic role. In the former problem the number of classes is 4; TARGET, I, B, and O. The number of classes in the latter problem depends on the number of semantic roles; given N semantic roles (filled with one or more words) and a single word TARGET, the number of classes is $2 * N + 1$. Both tasks can be viewed as multi-class classification problems. The first problem is the detection of the target word along with the segmentation of constituents into semantic chunks. On the other hand, the second problem is a harder problem of detecting the target word, segmenting constituents into chunks andlabeling each chunk with its corresponding semantic role.

We use SVMs for classification. However, they are binary classifiers. It is well known that a multi-class problem can be reduced to a number of binary-class problems. There are several approaches for extending SVMs to multiclass classification problems (Allwein et al., 2000). In this paper, we explore two approaches. The first approach is the extension of SVMs to multi-class classification using *codes*. Each class is assigned a codeword of 1's an -1's of length m. Here, m can be selected equal or greater than the number of classes. This splits the multi-class data into m binary class data. Therefore, one can design m SVM classifiers, combine their outputs and minimize a loss function to predict multi-class labels. One-vs-all, random and error correcting codes can be used for this purpose. The second approach is the creation of $\frac{N(N-1)}{2}$ binary problems by considering all possible pairs of classes, where N is the number of labels in the multi-class problem. This approach is called the pairwise classification. Some experiments have been reported that the pairwise approach outperforms the one-vs-all approach (Krebel, 1999).

The next step after the choice of classification method is the selection of features for classification. As done in (Kudo and Matsuamto, 2000), we derive features from a context that surrounds each word to be tagged. The features can be the words themselves, their POS tags, their positions in syntactic phrases and semantic roles that have already been classified. Figure 1 clarifies the notion of context and illustrates the features used. The 5-word sliding window centered at the current word defines the set of features.

In figure 2, we illustrate two possible implementations for feature extraction. However, at the moment, we don't know which one is better in terms of coverage, efficiency and accuracy.

Table 1: Target word detection and semantic role segmentation precision and recall rates with one-vs-all SVM classifiers.

Class	precision	recall	$F_{\beta=1}$
Phrase	66.1%	61.0%	63.5%
Target	79.5%	73.2%	76.2%

3 Experiments

We carried out experiments with a small fraction of the FrameNet database. A total of 3000 and 750 sentences are used for training and testing, respectively. All the experiments are carried out using TinySVM [1] and the general purpose SVM based chunker YamCha [2]. Table 1 summarizes the results for the simpler semantic chunking task using one-vs-all approach. The results corresponding to the pairwise approach are also presented in Table 2. Despite the claim in (Krebel, 1999) the pairwise approach has shown slightly inferior performance. To the best of our knowlege these are the first results on this type of task.

Table 2: Target word detection and semantic role segmentation precision and recall rates with pairwise SVM classifiers

Class	precision	recall	$F_{\beta=1}$
Phrase	65.3%	61.4%	63.3%
Target	76.8%	73.1%	74.9%

Table 3 shows the results for the second task. The precision and recall rates for all semantic roles are shown. The overall precision and recall rates are 67.6% and 55.9%, respectively. Comparing these to the respective rates of 65.0% and 61.0% in (Gildea and Jurafsky, 2002), in which the target word is assumed known, we find our very first results very promising and encouraging. Since FrameNet annotates the part of each sentence relevant to a particular target word in question, it is not optimal for training this type of classifier which must segment and classify entire sentences. We believe that the performance of the system is significantly degraded by this fact. We will be training on the PropBank corpus, which annotates all targets in a sentence and therefore be more appropriate for this type of mechanism.

4 Summary and Future Work

We have considered the automatic semantic role labeling as a chunking task and sucessfully ported a SVM based general chunker to our semantic chunking problem with encouraging results. As a future work, we plan to extend this work in several directions; (i) add new features (e.g. named entities, head words), (ii) use thresholding (e.g. based on information gain) to reduce the dimension of the feature space (iii) do both forward and backward parsing, and combine their results, and (iv) apply error correcting codes to improve the classification performance.

Table 3: Target word detection and semantic role segmentation and identification performances with one-vs-all SVM classifiers

Class	precision	recall
Target	79.9%	70.1%
agent	71.6%	65.9%
agent's_body_part	80.0%	40.0%
body_part	74.1%	77.4%
cause	44.4%	18.2%
degree	42.9%	42.9%
experiencer	52.9%	43.0%
force	40.0%	34.8%
goal	52.9%	43.9%
internal_cause	84.2%	50.0%
location	50.0%	36.1%
manner	56.9%	41.1%
path	61.7%	42.0%
patient	43.5%	33.8%
proposition	60.0%	29.3%
source	50.0%	25.0%
topic	69.2%	42.9%
undifined_fe	50.0%	14.3%
overall	67.6%	55.9%

References

Erin L. Allwein, Robert E. Schapire, and Yoram Singer. 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113-141.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkley FrameNet project. In *Proceedings of COLING/ACL*, pages 86-90.

C. J. Burges 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*. 2(2), pages 121-167, 1998.

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

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

Ulrich H.-G Krebel 1999 Pairwise classification and support vector machines. In *Advances in Kernel Methods*.

Taku Kudo, Yuji Matsumato 2000 Use of support vector learning for chunk identification. In *Proceedings of the 4th Conference on Very Large corpora*, pages 142-144.

Erik F. T. J. Sang, John Veenstra 1999. Representing text chunks. *Proceedings of EACL'99*, pages 173-179.

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

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