Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling

Xavier Carreras and **Lluís Màrquez**
TALP Research Centre
Technical University of Catalonia (UPC)
{carreras,lluism}@lsi.upc.es

Abstract

In this paper we describe the CoNLL-2004 shared task: semantic role labeling. We introduce the specification and goal of the task, describe the data sets and evaluation methods, and present a general overview of the systems that have contributed to the task, providing comparative description.

1 Introduction

In recent years there has been an increasing interest in semantic parsing of natural language, which is becoming a key issue in Information Extraction, Question Answering, Summarization, and, in general, in all NLP applications requiring some kind of semantic interpretation.

The shared task of CoNLL-2004 [1] concerns the recognition of semantic roles, for the English language. We will refer to it as *Semantic Role Labeling* (SRL). Given a sentence, the task consists of analyzing the propositions expressed by some target verbs of the sentence. In particular, for each target verb all the constituents in the sentence which fill a semantic role of the verb have to be extracted (see Figure 1 for a detailed example). Typical semantic arguments include Agent, Patient, Instrument, etc. and also adjuncts such as Locative, Temporal, Manner, Cause, etc.

Most existing systems for automatic semantic role labeling make use of a full syntactic parse of the sentence in order to define argument boundaries and to extract relevant information for training classifiers to disambiguate between role labels. Thus, the task has been usually approached as a two phase procedure consisting of *recognition* and *labeling* of arguments.

[1] CoNLL-2004 Shared Task web page —with data, software and systems' outputs available— at http://cnts.uia.ac.be/conll2004/roles.

Regarding the learning component of the systems, we find pure probabilistic models (Gildea and Jurafsky, 2002; Gildea and Palmer, 2002; Gildea and Hockenmaier, 2003), Maximum Entropy (Fleischman et al., 2003), generative models (Thompson et al., 2003), Decision Trees (Surdeanu et al., 2003; Chen and Rambow, 2003), and Support Vector Machines (Hacioglu and Ward, 2003; Pradhan et al., 2003a; Pradhan et al., 2003b).

There have also been some attempts at relaxing the necessity of using syntactic information derived from full parse trees. For instance, in (Pradhan et al., 2003a; Hacioglu and Ward, 2003), a SVM-based SRL system is devised which performs an IOB sequence tagging using only shallow syntactic information at the level of phrase chunks.

Nowadays, there exist two main English corpora with semantic annotations from which to train SRL systems: PropBank (Palmer et al., 2004) and FrameNet (Fillmore et al., 2001). In the CoNLL-2004 shared task we concentrate on the PropBank corpus, which is the Penn Treebank corpus enriched with predicate–argument structures. It addresses predicates expressed by verbs and labels core arguments with consecutive numbers (A0 to A5), trying to maintain coherence along different predicates. A number of adjuncts, derived from the Treebank functional tags, are also included in PropBank annotations.

To date, the best results reported on the PropBank correspond to a F_1 measure slightly over 83, when using the gold standard parse trees from Penn Treebank as the main source of information (Pradhan et al., 2003b). This performance drops to 77 when a real parser is used instead. Comparatively, the best SRL system based solely on shallow syntactic information (Pradhan et al., 2003a) performs more than 15 points below. Although these results are not directly comparable to the ones obtained in the CoNLL-2004 shared task (different datasets, different version of PropBank, etc.) they give an idea about the state-of-the art results on the task.

The challenge for CoNLL-2004 shared task is to come up with machine learning strategies which address the SRL problem on the basis of only partial syntactic information, avoiding the use of full parsers and external lexico-semantic knowledge bases. The annotations provided for the development of systems include, apart from the argument boundaries and role labels, the levels of processing treated in the previous editions of the CoNLL shared task, i.e., words, PoS tags, base chunks, clauses, and named entities.

The rest of the paper is organized as follows. Section 2 describes the general setting of the task. Section 3 provides a detailed description of training, development and test data. Participant systems are described and compared in section 4. In particular, information about learning techniques, SRL strategies, and feature development is provided, together with performance results on the development and test sets. Finally, section 5 concludes.

2 Task Description

The goal of the task is to develop a machine learning system to recognize arguments of verbs in a sentence, and label them with their semantic role. A verb and its set of arguments form a *proposition* in the sentence, and typically, a sentence will contain a number of propositions.

There are two properties that characterize the structure of the arguments in a proposition. First, arguments do not overlap, and are organized sequentially. Second, an argument may appear split into a number of non-contiguous phrases. For instance, in the sentence "[$_{\texttt{A1}}$ The apple], said John, [$_{\texttt{C–A1}}$ is on the table]", the utterance argument (labeled with type `A1`) appears split into two phrases. Thus, there is a set of non-overlapping arguments labeled with semantic roles associated with each proposition. The set of arguments of a proposition can be seen as a chunking of the sentence, in which chunks are parts of the semantic roles of the proposition predicate.

In practice, number of *target verbs* are marked in a sentence, each governing one proposition. A system has to recognize and label the arguments of each target verb.

2.1 Methodological Setting

Training and development data are provided to build the learning system. Apart from the correct output, both data sets contain the correct input, as well as predictions of the input made by state-of-the-art processors. The training set is used for training systems, whereas the development set is used to tune parameters of the learning systems and select the best model.

Systems have to be developed strictly with the data provided, which consists of input and output data and the official external resources (described below). Since the correct annotations for the input data are provided, a system is allowed either to be trained to predict the input part, or to make use of an external tool developed strictly within this setting, such as previous CoNLL shared task systems.

2.2 Evaluation

Evaluation is performed on a separate test set, which includes only predicted input data. A system is evaluated with respect to *precision*, *recall* and the F_1 measure. *Precision* (p) is the proportion of arguments predicted by a system which are correct. *Recall* (r) is the proportion of correct arguments which are predicted by a system. Finally, the F_1 measure computes the harmonic mean of precision and recall, and is the final measure to compare the performance of systems. It is formulated as: $\mathrm{F}_{\beta=1} = 2pr/(p+r)$.

For an argument to be correctly recognized, the words spanning the argument as well as its semantic role have to be correct. [2]

As an exceptional case, the verb argument of each proposition is excluded from the evaluation. This argument is the lexicalization of the predicate of the proposition. Most of the time, the verb corresponds to the target verb of the proposition, which is provided as input, and only in few cases the verb participant spans more words than the target verb.

Except for non-trivial cases, this situation makes the verb fairly easy to identify and, since there is one verb with each proposition, evaluating its recognition overestimates the overall performance of a system. For this reason, the verb argument is excluded from evaluation.

3 Data

The data consists of six sections of the Wall Street Journal part of the Penn Treebank (Marcus et al., 1993), and follows the setting of past editions of the CoNLL shared task: training set (sections 15-18), development set (section 20) and test set (section 21). We first describe annotations related to argument structure. Then, we describe the preprocessing of input data. Finally, we describe the format of the data sets.

3.1 PropBank

The Proposition Bank (PropBank) (Palmer et al., 2004) annotates the Penn Treebank with verb argument structure. The semantic roles covered by PropBank are the following:

- **Numbered arguments (`A0`–`A5`, `AA`):** Arguments defining verb-specific roles. Their semantics depends on the verb and the verb usage in a sentence, or *verb sense*. In general, `A0` stands for the *agent*

[2] The `srl-eval.pl` program is the official program to evaluate the performance of a system. It is available at the Shared Task web page.

and A1 corresponds to the *patient* or *theme* of the proposition, and these two are the most frequent roles. However, no consistent generalization can be made across different verbs or different senses of the same verb. PropBank takes the definition of verb senses from VerbNet, and for each verb and each sense defines the set of possible roles for that verb usage, called the *roleset*. The definition of rolesets is provided in the PropBank *Frames files*, which is made available for the shared task as an *official resource* to develop systems.

- **Adjuncts (AM-):** General arguments that any verb may take optionally. There are 13 types of adjuncts:

AM-ADV : general-purpose	AM-MOD : modal verb
AM-CAU : cause	AM-NEG : negation marker
AM-DIR : direction	AM-PNC : purpose
AM-DIS : discourse marker	AM-PRD : predication
AM-EXT : extent	AM-REC : reciprocal
AM-LOC : location	AM-TMP : temporal
AM-MNR : manner	

- **References (R-):** Arguments representing arguments realized in other parts of the sentence. The role of a reference is the same as the role of the referenced argument. The label is an R- tag prefixed to the label of the referent, e.g. R-A1.

- **Verbs (V):** Participant realizing the verb of the proposition, with exactly one verb for each one.

We used the February 2004 release of PropBank. Most predicative verbs were annotated, although not all of them (for example, most of the occurrences of the verb "to have" and "to be" were not annotated). We applied procedures to check consistency of propositions, looking for overlapping arguments, and incorrect semantic role labels. Also, co-referenced arguments were annotated as a single item in PropBank, and we automatically distinguished between the referent and the reference with simple rules matching pronominal expressions, which were tagged as R arguments. A total number of 68 propositions were not compliant with our procedures, and were filtered out from the CoNLL data sets. The predicate-argument annotations, thus, are not necessarily complete in a sentence. Table 1 provides counts of the number of sentences, annotated propositions, distinct verbs and arguments in the three data sets.

3.2 Preprocessing

In this section we describe the pipeline of processors to compute the annotations which form the input part of the data: part-of-speech (PoS) tags, chunks, clauses and named entities. The preprocessors correspond to the following state-of-the-art systems for each level of annotation:

	Training	Devel.	Test
Sentences	8,936	2,012	1,671
Tokens	211,727	47,377	40,039
Propositions	19,098	4,305	3,627
Distinct Verbs	1,838	978	855
All Arguments	50,182	11,121	9,598
A0	12,709	2,875	2,579
A1	18,046	4,064	3,429
A2	4,223	954	714
A3	784	149	150
A4	626	147	50
A5	14	4	2
AA	5	0	0
AM-ADV	1,727	352	307
AM-CAU	283	53	49
AM-DIR	231	60	50
AM-DIS	1,077	204	213
AM-EXT	152	49	14
AM-LOC	1,279	230	228
AM-MNR	1,337	334	255
AM-MOD	1,753	389	337
AM-NEG	687	131	127
AM-PNC	446	100	85
AM-PRD	10	3	3
AM-REC	2	1	0
AM-TMP	3,567	759	747
R-A0	738	162	159
R-A1	360	74	70
R-A2	49	17	9
R-A3	8	0	1
R-AA	1	0	0
R-AM-ADV	1	0	0
R-AM-LOC	27	4	4
R-AM-MNR	4	0	1
R-AM-PNC	1	0	1
R-AM-TMP	35	6	14

Table 1: Counts on the three data sets.

- PoS tagger: (Giménez and Màrquez, 2003), based on Support Vector Machines, and trained on Penn Treebank sections 0–18.

- Chunker and Clause Recognizer: (Carreras and Màrquez, 2003), based on Voted Perceptrons, and following the CoNLL settings of 2000 and 2001 tasks (Tjong Kim Sang and Buchholz, 2000; Tjong Kim Sang and Déjean, 2001). These two processors form a coherent partial syntax of a sentence, that is, chunks and clauses form a tree.

- Named entities with (Chieu and Ng, 2003), based on Maximum-Entropy classifiers, and following the CoNLL-2003 task setting (Tjong Kim Sang and De Meulder, 2003).

	Precision	Recall	F_1/Acc.
PoS Dev. (acc.)	–	–	96.88
PoS Test (acc.)	–	–	96.70
Chunking Dev.	94.28%	93.65%	93.96
Chunking Test	93.80%	92.93%	93.36
Clauses Dev.	90.51%	86.12%	88.26
Clauses Test	88.73%	82.92%	85.73
Named Entities	88.12%	88.51%	88.31

Table 2: Results of the preprocessing modules on the development and test sets. Named Entity figures are based on the CoNLL-2003 test set.

Such processors were ran in a pipeline, from PoS tags, to chunks, clauses and finally named entities. Table 2 summarizes the performance of the processors on the development and test sections. These figures differ from the original results in the original due to a better quality of the input information in our runs. The figures of the named entity extractor are based on the corpus of the CoNLL-2003 shared task, since gold annotations of named entities were not available for the current corpus.

3.3 Format

Figure 1 shows an example of a fully-annotated sentence. Annotations of a sentence are given using a flat representation in columns, separated by spaces. Each column encodes an annotation by associating a tag with every word. For each sentence, the following columns are provided:

1. Words.
2. Part of Speech tags.
3. Chunks in IOB2 format.
4. Clauses in Start-End format.
5. Named Entities in IOB2 format.
6. Target verbs, marking n predicative verbs. This column, provided as input, specifies the governing verbs of the propositions to be analyzed. Each target verb is in the base form. Occasionally this column does not mark any verb (i.e., n may be 0).
7. For each of the n target verbs, a column in Start-End format specifying the arguments of the proposition. These columns are the output of a system, that is, the ones to be predicted, and are not available for the test set.

IOB2 format. Represents chunks which do not overlap nor embed. Words outside a chunk receive the tag O. For words forming a chunk of type k, the first word receives the `B-`k tag (Begin), and the remaining words receive the tag `I-`k (Inside).

Start-End format. Represents non-overlapping phrases (clauses or arguments) which may be embedded[3] inside one another. Each tag indicates whether a clause starts or ends at that word and is of the form `START*END`. The `START` part is a concatenation of `(`k parentheses, each representing that a phrase of type k starts at that word. The `END` part is a concatenation of k`)` parentheses, each representing that a phrase of type k ends at that word. For example, the `*` tag represents a word with no starts and ends; the `(A0*A0)` tag represents a word constituting an `A0` argument; and the `(S(S*S)` tag represents a word which constitutes a base clause (labeled `S`) and starts another higher-level clause. Finally, the concatenation of all tags constitutes a well-formed bracketing. For the particular case of split arguments, of type k, the first part appears as a phrase with label k, and the remaining as phrases with label `C-`k (continuation prefix). See examples of annotations at columns 4th, 7th and 8th of Figure 1.

4 Participating Systems

Ten systems have participated in the CoNLL-2004 shared task. They approached the task in several ways, using different learning components and labeling strategies. The following subsections briefly summarize the most important properties of each system and provide a qualitative comparison between them, together with a quantitative evaluation on the development and test sets.

4.1 Learning techniques

Up to six different learning algorithms have been applied in the CoNLL-2004 shared task. None of them is new with respect to the past editions. Two teams used the Maximum Entropy (ME) statistical framework (Baldewein et al., 2004; Lim et al., 2004). Two teams used Brill's Transformation-based Error-driven Learning (TBL) (Higgins, 2004; Williams et al., 2004). Two other groups applied Memory-Based Learning (MBL) (van den Bosch et al., 2004; Kouchnir, 2004). The remaining four teams employed vector-based linear classifiers of different types: Hacioglu et al. (2004) and Park et al. (2004) used Support Vector Machines (SVM) with polynomial kernels, Carreras et al. (2004) used Voted Perceptrons (VP) also with polynomial kernels, and finally, Punyakanok et al. (2004) used SNoW, a Winnow-based network of linear separators. Additionally, the team of Baldewein et al. (2004) used a EM–based clustering algorithm for feature development (see section 4.3).

As a main difference with respect to past editions, less effort has been put into combining different learning algorithms and outputs. Instead, the main effort of participants went into developing useful SRL strategies and into the development of features (see sections 4.2 and 4.3). As an exception, van den Bosch et al. (2004) applied a

[3] Arguments in data do not embed, though format allows so.

```
The          DT   B-NP    (S*      O      -          (A0*                *
San          NNP  I-NP      *      B-ORG  -             *                *
Francisco    NNP  I-NP      *      I-ORG  -             *                *
Examiner     NNP  I-NP      *      I-ORG  -             *A0)             *
issued       VBD  B-VP      *      O      issue       (V*V)              *
a            DT   B-NP      *      O      -          (A1*             (A1*
special      JJ   I-NP      *      O      -             *                *
edition      NN   I-NP      *      O      -             *A1)             *A1)
around       IN   B-PP      *      O      -      (AM-TMP*                *
noon         NN   B-NP      *      O      -             *AM-TMP)         *
yesterday    NN   B-NP      *      O      -      (AM-TMP*AM-TMP)         *
that         WDT  B-NP    (S*      O      -        (C-A1*           (R-A1*R-A1)
was          VBD  B-VP    (S*      O      -             *                *
filled       VBN  I-VP      *      O      fill          *              (V*V)
entirely     RB   B-ADVP    *      O      -             *        (AM-MNR*AM-MNR)
with         IN   B-PP      *      O      -             *                *
earthquake   NN   B-NP      *      O      -             *             (A2*
news         NN   I-NP      *      O      -             *                *
and          CC   I-NP      *      O      -             *                *
information  NN   I-NP      *S)S)  O      -             *C-A1)           *A2)
.            .    O         *S)    O      -             *                *
```

Figure 1: An example of an annotated sentence, in columns. Input consists of words (1st), PoS tags (2nd), base chunks (3rd), clauses (4th) and named entities (5th). The 6th column marks target verbs, and their propositions are found in remaining columns. According to the PropBank Frames, for `issue` (7th), the `A0` annotates the issuer, and the `A1` the thing issued, which appears split into two parts. For `fill` (8th), `A1` is the the destination, and `A2` the theme.

voting strategy to derive the final sequence tagging as a voted combination of three overlapping n-gram output sequences. The same team also applied a meta-learning step, by using iterative classifier stacking, for correcting systematic errors committed by the low–level classifiers. This work is also worth mentioning because of the extensive work done on parameter tuning and feature selection.

4.2 SRL approaches

SRL is a complex task which has to be decomposed into a number of simpler decisions and tagging schemes in order to be addressed by learning techniques.

One first issue is the annotation of the different propositions of a sentence. Most of the groups treated the annotation of semantic roles for each verb predicate as an independent problem. An exception is the system of Carreras et al. (2004), which performs the annotation of all propositions simultaneously. As a consequence, the former teams treat the problem as the recognition of sequential structures (a.k.a. chunking), while the latter directly derives a hierarchical structure formed by the arguments of all propositions. Table 3 summarizes the main properties of each system regarding the SRL strategy implemented. This property corresponds to the first column.

Regarding the *labeling strategy*, we can distinguish at least three different strategies. The first one consists of performing role identification directly by a IOB-type sequence tagging. The second approach consists of dividing the problem into two independent phases: *recognition*, in which the arguments are recognized, and *labeling*, in which the already recognized arguments are assigned role labels. The third approach also proceeds in two phases: *filtering*, in which a set of argument candidates are decided and *labeling*, in which the set of optimal arguments is derived from the proposed candidates. As a variant of the first two-phase strategy, van den Bosch et al. (2004) first perform a direct classification of chunks into argument labels, and then decide the actual arguments in a post-process by joining previously classified argument fragments. All this information is summarized in the second column of Table 3.

An implication of implementing the two-phase strategy is the ability to work with argument candidates in the second phase, allowing to develop feature patterns for complete arguments. Regarding the first phase, the recognition of candidate arguments is performed by means of a IOB or *open–close* tagging using classifiers, either argument–independent, or specialized by argument type.

It is also worth noting that all participant systems performed learning of predicate-independent classifiers instead of specializing by the verb predicate. Information about verb predicates is captured through features and some global restrictions.

Another important issue is the *granularity* at which the sentence elements are processed. It has become very clear that a good election for this problem is phrase-by-phrase processing (P-by-P, using the notation introduced by Hacioglu et al. (2004)) instead of word-by-word (W-by-W). The motivation is twofold: (1) phrase boundaries are almost always consistent with argument boundaries;

(2) P-by-P processing is computationally less expensive and allows to explore a relatively larger context. Most of the groups performed a P-by-P processing, but admitting a processing by words within the target verb chunks. The system by Baldewein et al. (2004) works with a bit more general elements called "chunk sequences", extracted in a preprocess using heuristic rules. This information is presented in the third column of Table 3.

Information regarding clauses has proven to be very useful, as can be seen in section 4.3. All systems captured some kind of clause information through feature codification. However, some of the systems restrict the search for arguments only to the immediate clause (Park et al., 2004; Williams et al., 2004) and others use the clause hierarchy to guide the exploration of the sentence (Lim et al., 2004; Carreras et al., 2004).

Very relevant to the SRL strategy is the availability of global sentential information when decisions are taken. Almost all of the systems try to capture some global level information by collecting features describing the target predicate and its context, the "syntactic path" from the element under consideration to the predicate, etc. (see section 4.3). But only some of them include a global optimization procedure at sentence level in the labeling strategy. The systems working with Maximum Entropy Models (Baldewein et al., 2004; Lim et al., 2004) use beam search to find taggings that maximize the probability of the output sequence. Carreras et al. (2004) and Punyakanok et al. (2004) also define a global scoring function to maximize. At this point, the system of Punyakanok et al. (2004) deserves special consideration, since it formally implements a set of structural and linguistic constraints directly in the global cost function to maximize. These constraints act as a filter for valid output sequences and ensure coherence of the output. Authors refer to this part of the system as the *inference layer* and they implement it using integer linear programming. The iterative classifier stacking mechanism used by van den Bosch et al. (2004) also tries to alleviate the problem of locality of the low-level classifiers. This information is found in the fourth column of Table 3.

Finally, some systems use some kind of postprocessing to ensure coherence of the final labeling, correct some systematic errors, or to treat some types of adjunctive arguments. In most of the cases, this postprocess is performed on the basis of simple ad-hoc rules. This information is included in the last column of Table 3.

	prop.	lab.	gran.	glob.	post
hacioglu	s	t	P-by-P	no	no
punyakanok	s	fl	W-by-W	yes	no
carreras	j	fl	P-by-P	yes	no
lim	s	t	P-by-P	yes	no
park	s	rc	P-by-P	no	yes
higgins	s	t	W-by-W	no	yes
van den bosch	s	cj	P-by-P	part.	yes
kouchnir	s	rc	P-by-P	no	yes
baldewein	s	rc	P-by-P	yes	no
williams	s	t	mixed	no	no

Table 3: Main properties of the SRL strategies implemented by the ten participant teams (sorted by performance on the test set). "prop." stands for the treatment of all propositions of a sentence; possible values are: `s` (separate) and `j` (joint). "lab." stands for labeling strategy; possible values are: `t` (one step tagging), `rc` (recognition + classification), `fl` (filtering + labeling), `cj` (classification + joining). "gran." stands for granularity; "glob." stands for global optimization. "post" stands for post-processing.

4.3 Features

With a very few exceptions all the participant systems have used all levels of linguistic information provided in the training data sets, that is, words, PoS and chunk labels, clauses, and named entities.

It is worth mentioning that the general type of features derived from the basic information are strongly inspired by previous works on the SRL task (Gildea and Jurafsky, 2002; Surdeanu et al., 2003; Pradhan et al., 2003a). Many systems used the same kind of ideas but implemented in different ways, since the particular learning strategies used (see section 4.2) impose different constraints on the type of information available or the way of expressing it.

As a general idea, we can divide the features into four types: (1) *basic* features, evaluating some kind of local information on the context of the word or constituent being treated; (2) Features characterizing the internal structure of a candidate argument; (3) Features describing properties of the target verb predicate; (4) Features that capture the relations between the verb predicate and the constituent under consideration.

All systems used some kind of basic features. Roughly speaking, they consist of words, PoS tags, chunks, clause labels, and named entities extracted from a window-based context. These values can be considered with or without the relative position with respect to the element under consideration, and some n-grams of them can also be computed. If the granularity of the system is at phrase level then typically a representative head word of the phrase is used as lexical information. As an exception to the general approach, the system of Williams et al. (2004) does not make use of word forms.

The rest of the features are more interesting since they are task dependent, and deserve special attention. Table 4 summarizes the type of features exploited by systems.

To represent an argument itself, few attributes are of

general usage. Some systems count the length of it, with different granularities. Others make use of heuristics to derive its syntactic type. There are systems that extract a structured representation of the argument, either homogeneous (capturing different sequences of head words, PoS tags, chunks or clauses), or heterogeneous (combining all elements, based on the syntactic hierarchy). A few systems have captured the existence of neighboring arguments, previously identified in the process. Interestingly, the system of Lim et al. (2004) represents the context of an argument relative to the syntactic hierarchy by means of relative constituent sequences and syntactic levels. Concerning lexicalization of the argument, most of the techniques rely on head word rules based on Collins', or content word rules as in Surdeanu et al. (2003). Only Carreras et al. (2004) decide to use a bag-of-words model, apart from heuristic-based lexicalization.

Regarding the target verb, the voice feature of the verb is generally used, in addition to basic features capturing the form and PoS tag of the verb. Some systems captured statistics on frequent argument patterns for each predicate. Also, systems represented the elements in the proximity of the target verb, inspired by local subcategorization patterns of a predicate.

As for features related to a constituent-predicate pair, all systems use the simple feature describing the relative position between them, and to a lesser degree, the distance and the difference in clausal levels. Again, there is a general tendency to describe the structured path from the argument to the verb. Its design goes from simple homogeneous sequences of head words or chunks, to more sophisticated paths combining chunks and clauses, and capturing hierarchical properties. The system of Park et al. (2004) also tracks the number of different syntactic elements found between the pair. Remarkably, the system of Baldewein et al. (2004) uses an EM clustering technique to derive features representing the affinity of an argument and a predicate.

On top of basic feature extraction, all teams working with SVM and VP used polynomial kernels of degree 2. Similar in expressiveness, the system designed by Punyakanok et al. (2004) expanded the feature space with all pairs of basic features.

4.4 Evaluation

A baseline rate was computed for the task. It was produced by a system developed by Erik Tjong Kim Sang, from the University of Antwerp, Belgium. The baseline processor finds semantic roles based on the following seven rules:

- Tag target verb and successive particles as `V`.
- Tag `not` and `n't` in target verb chunk as `AM-NEG`.
- Tag modal verbs in target verb chunk as `AM-MOD`.
- Tag first `NP` before target verb as `A0`.
- Tag first `NP` after target verb as `A1`.
- Tag `that`, `which` and `who` before target verb as `R-A0`.
- Switch `A0` and `A1`, and `R-A0` and `R-A1` if the target verb is part of a passive `VP` chunk. A `VP` chunk is considered in passive voice if it contains a form of `to be` and the verb does not end in `ing`.

Table 5 presents the overall results obtained by the ten participating systems, on the development and test sets. The best performance was obtained by the SVM-based IOB tagger of (Hacioglu et al., 2004), which almost reached the performance of 70 in F_1 on the test. The seven best systems obtained F_1 scores in the range of 60-70, and only three systems scored below that.

Comparing the results across development and test corpora, most systems experienced a decrease in performance between 1.5 and 3 points. As in previous editions of the shared task, we attribute this behavior to a greater difficulty of the test set instead of an overfitting effect. Interestingly, the three systems performing below 60 in the development set did not experienced this decrease. In fact (Williams et al., 2004) and (Baldewein et al., 2004) even improved the results on the test set.

Table 6 details the performance of systems for the A0-A4 arguments, on the test set. Consistently, the best performing system of the task also outperforms all other systems on these semantic roles.

5 Conclusion

We have described the CoNLL-2004 shared task on semantic role labeling. The task was based on the PropBank corpus, and the challenge was to come up with machine learning techniques to recognize and label semantic roles on the basis of partial syntactic structure. Ten systems have participated to the task, contributing with a variety of standard or novel learning architectures. The best system, presented by the most experienced group on the task (Hacioglu et al., 2004), achieved a moderate performance of 69.49 at the F_1 measure. It is based on a SVM tagging system, performing IOB decisions on the chunks of the sentence, and exploiting a wide variety of features based on partial syntax.

Most of the systems advance the state-of-the-art on semantic role labeling on the basis of partial syntax. However, state-of-the-art systems working with full syntax still perform substantially better, although far from a desired behavior for real-task application. Two questions remain open: which syntactic structures are needed as input for the task, and what other sources of information are required to obtain a real-world, accurate performance.

As a future line, a more thorough experimental evaluation is required to see which are the components that

	sy	ne	al	at	as	aw	an	vv	vs	vf	vc	rp	di	pa	ex
hacioglu	+	+	+	−	−	+	−	+	+	−	+	+	+	+	+
punyakanok	+	+	+	+	+	+	−	+	−	+	+	+	−	+	+
carreras	+	−	−	−	+	+	−	+	−	−	−	+	−	+	+
lim	+	−	−	−	−	+	+	+	−	−	−	+	−	+	−
park	+	−	−	−	−	−	−	+	−	−	+	+	+	+	+
higgins	+	+	−	−	−	−	+	+	−	−	−	+	+	+	−
van den bosch	+	+	−	−	−	−	−	+	+	−	−	+	+	−	−
kouchnir	+	−	+	−	+	+	−	+	−	+	−	+	+	−	−
baldewein	+	+	+	+	+	+	−	+	+	−	−	+	+	−	−
williams	+	+	−	−	−	−	−	−	−	−	−	+	−	−	−

Table 4: Main feature types used by the 10 participating systems in the CoNLL-2004 shared task, sorted by performance on the test set. "sy": use of partial syntax (all levels); "ne": use of named entities; "al": argument length; "at": argument type; "as": argument internal structure; "aw": head-word lexicalization of arguments; "an": neighboring arguments; "vv": verb voice; "vs": verb statistics; "vf": verb features derived from PropBank frames; "vc": verb local context; "rp": relative position; "di": distance (horizontal or in the hierarchy); "pa": path; "ex": feature expansion.

most contributed to the performance of systems.

Acknowledgements

Authors would like to thank the following people and institutions. The PropBank team, and specially Martha Palmer and Scott Cotton, for making the corpus available. The CoNLL-2004 board for fruitful discussions and suggestions. In particular, Erik Tjong Kim Sang for useful comments from his valuable experience, and for making the baseline SRL processor available. Lluís Padró and Mihai Surdeanu, Grzegorz Chrupała, and Hwee Tou Ng for helping us in the reviewing process and the preparation of this document. Finally, the teams contributing to shared task, for their great interest in participating.

This work has been partially funded by the European Commission (Meaning, IST-2001-34460) and the Spanish Research Department (Aliado, TIC2002-04447-C02). Xavier Carreras is supported by a pre-doctoral grant from the Catalan Research Department.

References

Ulrike Baldewein, Katrin Erk, Sebastian Padó, and Detlef Prescher. 2004. Semantic role labeling with chunk sequences. In *Proceedings of CoNLL-2004*.

Xavier Carreras and Lluís Màrquez. 2003. Phrase recognition by filtering and ranking with perceptrons. In *Proceedings of RANLP-2003*, Borovets, Bulgaria.

Xavier Carreras, Lluís Màrquez, and Grzegorz Chrupała. 2004. Hierarchical recognition of propositional arguments with perceptrons. In *Proceedings of CoNLL-2004*.

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.

Hai Leong Chieu and Hwee Tou Ng. 2003. Named entity recognition with a maximum entropy approach. In *Proceedings of CoNLL-2003*, Edmonton, Canada.

Charles J. Fillmore, Charles Wooters, and Collin F. Baker. 2001. Building a large lexical databank which provides deep semantics. In *Proceedings of the Pacific Asian Conference on Language, Informa tion and Computation*, Hong Kong, China.

Michael Fleischman, Namhee Kwon, and Eduard Hovy. 2003. Maximum entropy models for framenet classification. In *Proceedings of EMNLP-2003*, Sapporo, Japan.

Daniel Gildea and Julia Hockenmaier. 2003. Identifying semantic roles using combinatory categorial grammar. In *Proceedings of EMNLP-2003*, Sapporo, Japan.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Daniel Gildea and Martha Palmer. 2002. The necessity of parsing for predicate argument recognition. In *Proceedings of ACL 2002*, Philadelphia, USA.

Jesús Giménez and Lluís Màrquez. 2003. Fast and accurate part-of-speech tagging: The svm approach revisited. In *Proceedings of RANLP-2003*, Borovets, Bulgaria.

Kadri Hacioglu and Wayne Ward. 2003. Target word detection and semantic role chunking using support vector machines. In *Proceedings of HLT-NAACL 2003*, Edmonton, Canada.

Kadri Hacioglu, Sameer Pradhan, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2004. Semantic role labeling by tagging syntactic chunks. In *Proceedings of CoNLL-2004*.

Derrick Higgins. 2004. A transformation-based approach to argument labeling. In *Proceedings of CoNLL-2004*.

development	Precision	Recall	F_1
hacioglu	74.18%	69.43%	71.72
punyakanok	71.96%	64.93%	68.26
carreras	73.40%	63.70%	68.21
lim	69.78%	62.57%	65.97
park	67.27%	64.36%	65.78
higgins	65.59%	60.16%	62.76
van den bosch	69.06%	57.84%	62.95
kouchnir	44.93%	63.12%	52.50
baldewein	64.90%	41.61%	50.71
williams	53.37%	32.43%	40.35
baseline	50.63%	30.30%	37.91

test	Precision	Recall	F_1
hacioglu	72.43%	66.77%	69.49
punyakanok	70.07%	63.07%	66.39
carreras	71.81%	61.11%	66.03
lim	68.42%	61.47%	64.76
park	65.63%	62.43%	63.99
higgins	64.17%	57.52%	60.66
van den bosch	67.12%	54.46%	60.13
kouchnir	56.86%	49.95%	53.18
baldewein	65.73%	42.60%	51.70
williams	58.08%	34.75%	43.48
baseline	54.60%	31.39%	39.87

Table 5: Overall precision, recall and F_1 rates obtained by the ten participating systems in the CoNLL-2004 shared task on the development and test sets.

	A0	A1	A2	A3	A4
hacioglu	81.37	71.63	49.33	51.11	66.67
punyakanok	79.38	68.16	46.69	34.04	65.22
carreras	79.05	66.96	43.28	31.22	62.07
lim	77.42	66.00	49.07	41.77	54.55
park	76.38	66.14	46.57	42.32	51.76
higgins	70.67	62.72	45.52	40.00	39.64
van den bosch	74.95	60.83	40.41	37.44	62.37
kouchnir	65.49	54.48	30.95	19.71	36.07
baldewein	66.76	53.37	37.60	22.89	27.69
williams	56.24	49.05	00.00	00.00	00.00
baseline	57.65	34.19	00.00	00.00	00.00

Table 6: F_1 scores on the most frequent core argument types obtained by the ten participating systems in the CoNLL-2004 shared task on the test set. Systems sorted by overall performance on the test set.

Beata Kouchnir. 2004. A memory-based approach for semantic role labeling. In *Proceedings of CoNLL-2004*.

Joon-Ho Lim, Young-Sook Hwang, So-Young Park, and Hae-Chang Rim. 2004. Semantic role labeling using maximum entropy model. In *Proceedings of CoNLL-2004*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2004. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*. Submitted.

Kyung-Mi Park, Young-Sook Hwang, and Hae-Chang Rim. 2004. Two-phase semantic role labeling based on support vector machines. In *Proceedings of CoNLL-2004*.

Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2003a. Support vector learning for semantic argument classification. Technical Report TR-CSLR-2003-03, Center for Spoken Language Research, University of Colorado.

Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2003b. Semantic role parsing: Adding semantic structure to unstructured text. In *Proceedings of the International Conference on Data Mining (ICDM-2003)*, Melbourne, USA.

Vasin Punyakanok, Dan Roth, Wen-Tau Yih, Dav Zimak, and Yuancheng Tu. 2004. Semantic role labeling via generalized inference over classifiers. In *Proceedings of CoNLL-2004*.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of ACL 2003*, Sapporo, Japan.

Cynthia A. Thompson, Roger Levy, and Christopher D. Manning. 2003. A generative model for semantic role labeling. In *Proceedings of ECML'03*, Dubrovnik, Croatia.

E. F. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of the 4th Conference on Natural Language Learning, CoNLL-2000*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*.

Erik F. Tjong Kim Sang and Hervé Déjean. 2001. Introduction to the CoNLL-2001 shared task: Clause identification. In *Proceedings of the 5th Conference on Natural Language Learning, CoNLL-2001*.

Antal van den Bosch, Sander Canisius, Walter Daelemans, Iris Hendrickx, and Erik Tjong Kim Sang. 2004. Memory-based semantic role labeling: Optimizing features, algorithm, and output. In *Proceedings of CoNLL-2004*.

Ken Williams, Christopher Dozier, and Andrew McCulloh. 2004. Learning transformation rules for semantic role labeling. In *Proceedings of CoNLL-2004*.