Generative and Discriminative Learning in Semantic Role Labeling for Italian

Cristina Giannone

Department of Enterprise Engineering
University of Roma, Tor Vergata
giannone@info.uniroma2.it

Abstract. In this paper, we present a Semantic Role Labeling tool for Italian language for the FLaIT competition at Evalita 2011. This tool presents an hybrid approach to resolve the different sub-tasks that composed the SRL task. We apply a discriminative model for the boundary detection task based on lexical and syntactical features. A distributional approach to modeling lexical semantic information, instead, for the Argument Classification sub-task is applied in a semi-supervised perspective. The combination of these models achieved interesting results in the FLaIT competition.

Keywords: Semantic role labeling, Framenet, distributional semantics, geometrical models, SVM

1 Introduction

In the Frame Labeling over Italian Texts (FLaIT) evaluation proposed in Evalita 2011 we present a system for the automatic labeling of semantic roles defined by the semantic theory FrameNet. The proposed tool performs the whole chain for the Semantic Role Labeling task, we did not participate, instead, to the *Frame Prediction* subtask. This SRL tool consists in three computational steps:

- Boundary Detection (BD): Identifying the boundaries of arguments of the lexical unit.
- Argument Classification (AC): Labeling the identified boundaries with the correct semantic roles, in an independent process for each boundary.
- Join Re-ranking (RR) : The joint model is used to decide the entire argument sequence among the set of the n-best competing solutions computed in the previous step.

The Evalita 2011 FLaIT challenge is the first tentative to evaluate SRL tools for Italian language. Up until now, only one work for Italian language has been proposed over a small dataset [2]. Although several machine learning models obtain interesting results, they present some limitations in term of generalization power with a consequential loss of labeling accuracy. As discussed also in [10, 6] this phenomena affects mainly the performance of argument classification (AC)

process in out-of-domain tests and in poor training conditions (e.g. over non English languages characterized by few annotated examples) is still significant. In the proposed tool, in order to overcame these limitations, we adopt two different learning approaches to train the different modules corresponding to the subtasks. In particular, we adopt a discriminative boundary detection model trained with lexical and syntactic features as in the study of [6]. The comprehensive list of features used in learning of BD models is discussed in Section 2. In Section 3 we focuses the discussion on the argument classification step. The model we present adopts a simple feature space by relying on a limited set of grammatical properties, thus reducing its learning capacity. Moreover, it generalizes lexical information about the annotated examples by applying a geometrical model, in a Latent Semantic Analysis style, inspired by a distributional paradigm [9], while Section 4 describes a joint re-ranking module based on HMM model. Finally, the results achieved in FLaIT and some final conclusions are discussed in Section 5.

2 Boundary Detection

In this experimentation we adopt a boundary detection model trained with lexical and syntactic features as in the study of [6]. We trained a boundary detection (BD) model using SVM classifier[1] . The features used are in line with the classical BD approaches described in [4]. We trained three different models, one for the most frequent part-of-speeches of lexical unit : *verbs, nouns, adjectives*. This choice was made in order to reduce the data sparseness in the feature space. Features used in BD training are discussed below. We distinguish them along two lines: syntactically and lexically based.

2.1 Syntactic Features

Syntactic features represent all the information coming from the dependency graph.

- **Part-of-speech** : Part of Speech of the following tokens: Lexical unit, Head argument, Rightmost dependent of the argument head, Leftmost dependent of the argument head, Parent node of the lexical unit.
- **Position** Position of the head word of the argument with respect to the lexical unit: Before, After, or On.
- **Voice** Define the form of the verbal lexical units (active or passive).
- **Dependency Path** A string representation of the path through the dependency parse from the target node to the argument node.
- **Relation to Parent** : Dependency relation between the lexical unit and its parent.
- **Parent Has Obj** : Feature that is set to true if the parent of the lexical unit has an object.

[1] In this experimentation we use the SVM svmLight software release. http://svmlight.joachims.org/

– **Grammatical Function** : The grammatical function of the argument node.
– **Child Dep Set** : The set of grammatical functions of the direct dependents of the lexical unit.

2.2 Lexical Features

The lexical features exploit the lexical level of the sentence.
– **Lemmas**: The following lemmas: Lexical unit, Frame element argument head, Rightmost dependent of the argument head, Leftmost dependent of the argument head, Parent node of the lexical unit.
– **FrameElements**: The list of the core frame elements for a given frame.

In this model, we adopt the dependency syntactic annotation provided in the FLaIT dataset, applying the classifier on dependency nodes. This approach could be affect by the parser errors in cases in witch an argument boundary does not fully match with the exact span of a dependency node.

3 Argument Classification

In the argument classification step, we have explored two different aspects. First, we propose a model that does not depend on complex syntactic information in order to minimize the risk of overfitting, generalizing lexical information about the annotated examples by applying a geometrical model, in a Latent Semantic Analysis style, inspired by a distributional paradigm [9]. Second, we improve the lexical semantic information available to the learning algorithm. The proposed "minimalistic" approach will consider only two independent features:

– the *semantic head* (h) of a role, as it can be observed in the grammatical structure.
– the *dependency relation* (r) connecting the semantic head to the predicate words.

In distributional models, words are thus represented through vectors built over these observable contexts: similar vectors suggest *semantic relatedness* as a function of the distance between two words, capturing paradigmatic (e.g. synonymy) or syntagmatic relations [8].

Vectors $\overrightarrow{h}$ are described by an adjacency matrix M, whose rows describe target words (h) and whose columns describe their corpus contexts. Latent Semantic Analysis (LSA) [7], is then applied to M to acquire meaningful representations $\overrightarrow{h}$ for individual heads h (i.e., the target words). LSA exploits the linear transformation called *Singular Value Decomposition* (SVD) and produces an approximation of the original matrix M, capturing (semantic) dependencies between context vectors.

In the argument classification task, the similarity between two argument heads h_1 and h_2 observed in FrameNet can be computed over $\overrightarrow{h_1}$ and $\overrightarrow{h_2}$. The

model for a given frame element FE^k is built around the semantic heads h observed in the role FE^k in the training set: they form a set denoted by H^{FE^k}.

These LSA vectors $\overrightarrow{h}$ express the individual annotated examples as they are immerse in the LSA space acquired from the unlabeled texts. Moreover, given FE^k, a model for each individual syntactic relation r (i.e. that links h labeled as FE^k to their corresponding predicates) is a partition of the set H^{FE^k} called $H_r^{FE^k}$, i.e. the subset of H^{FE^k} produced by examples of the relation r (e.g. `Subj`).

As the LSA vectors $\overrightarrow{h}$ are available for the semantic heads h, a vector representation $\overrightarrow{FE^k}$ for the role FE^k can be obtained from the annotated data. However, one single vector is a too simplistic representation given the rich nature of semantic roles FE^k. In order to better represent FE^k, multiple regions in the semantic space are used. They are obtained by a clustering process applied to the set $H_r^{FE^k}$ according to the *Quality Threshold (QT)* algorithm [5].

For a frame F, clusters define a geometric model of every frame elements FE^k: it consists of centroids $\overrightarrow{c}$ with $c \subseteq H_r^{FE^k}$. Each c represents FE^k through a set of similar heads, as role fillers observed in FrameNet. A sentence s can be seen as a sequence of role-relation pairs: $s = \{(r_1, h_1),\ ..., (r_n, h_n)\}$ where the heads h_i are in the syntactic relation r_i with the underlying lexical unit of F.

For every head h in s, the vector $\overrightarrow{h}$ can be then used to estimate its similarity with the different candidate roles FE^k. Given the syntactic relation r, the clusters $c \in C_r^{FE^k}$ whose centroid vector $\boldsymbol{c}$ is closer to $\boldsymbol{h}$ are selected.

In some cases information about the head h is not available from the unlabeled corpus or no example of relation r for the role FE^k is available from the annotated corpus. Often the incoming head h or the relation r may be unavailable:

1. If the head h has never been met in the unlabeled corpus or the high grammatical ambiguity of the sentence does not allow to locate it reliably, the distributional model should be backed off to a purely syntactic model, that is $prob(FE^k|r)$
2. If the relation r can not be properly located in s, h is also unknown: the prior probability of individual arguments, i.e. $prob(FE^k)$, is here employed.

Both $prob(FE^k|r)$ and $prob(FE^k)$ can be estimated from the training set and smoothing can be also applied[2]. A more robust argument preference function for all arguments $(r_i, h_i) \in s$ of the frame F is thus given by:

$$prob(FE^k|r_i, h_i) = \lambda_1 prob(FE^k|r_i, h_i) + \lambda_2 prob(FE^k|r_i) + \lambda_3 prob(FE^k) \quad (1)$$

where weights λ_1, λ_2, λ_3 can be heuristically assigned or estimated from the training set[3]. The resulting model is called *Backoff model*: although simply based on a single feature (i.e. the syntactic relation r), it accounts for information at different reliability degrees.

[2] Lindstone smoothing was applied with $\delta = 1$.

[3] In each test discussed hereafter, λ_1, λ_2, λ_3 were assigned to .9,.09 and .01, in order to impose a strict priority to the model contributions.

4 Join Re-ranking

Eq. 1 defines roles preferences local to individual arguments (r_i, h_i). However, an argument frame is a joint structure, with strong dependencies between arguments. We thus propose to model the re-ranking phase (RR) as a HMM sequence labeling task. It defines a stochastic inference over multiple (locally justified) alternative sequences through a Hidden Markov Model (HMM). It infers the best sequence $FE^{(k_1,\ldots,k_n)}$ over all the possible hidden state sequences (i.e. made by the target FE^{k_i}) given the observable emissions, i.e. the arguments (r_i, h_i). Viterbi inference is applied to build the best (role) interpretation for the input sentence.

5 Experimental Evaluation and Conclusion

The dataset provided by the Flait team is composed by 1255 annotated sentences for the training set and 318 for the test one. We use 5-fold cross validation to assess the model parameters. For the Boundary Classification subtask we make use only the annotated dataset provided by the organization team. In the Argument Classification task, instead, an external unannotated corpus, the ItWaC [1], is used to compute the LSA space. The ItWaC corpus is composed by approximately 2 billion word collection of written Italian from the web, from an unknown variety of genres. The entire ItWaC corpus has been parsed and the dependency graphs derived from individual sentences provided the basic observable contexts: every co-occurrence is thus syntactically justified by a dependency arc. The most frequent 20,000 basic features, i.e. (syntactic relation,lemma) pairs, have been used to build the matrix M, vector components corresponding to point-wise mutual information scores. Finally, the final space is obtained by applying the SVD reduction over M, with a dimensionality cut of $l = 250$. The proposed SRL tool participates to the second and third test run. In the second test run the boundary recognition is evaluated. The test set is provided with explicit information about the correct frame corresponding to the marked lexical unit. The resulting boundary detection accuracy achieves the state-of-the-art in the token based evaluation and is the second best result in the perfect match evaluation. As previously discussed, this is mainly due to errors in the dependency parsing. The argument classification accuracy in the second run is calculated over the boundaries tagged by the tool, performing the full SRL chain. In this sub-task our model achieves the second best results in the competition. Misclassified arguments are caused by the not availability of some heads in the distributional space. The backoff model in some cases is not able to find the correct argument due the estimation over a small size training dataset provided for some frames.

In the third run the test are released with the explicit information about the marked boundaries of individual arguments. Here the AC task is required. In this run our tool achieves interesting results achieving a F1 of 65.82% in the perfect matching. Preliminary error analysis confirms the previous consideration about

Table 1. Results of the second and third runs : Boundary Detection (BD) and Argument Classification (AC)

Task	Precision(%)	Recall (%)	F1 (%)	Token based Precision(%)	Token based Recall(%)	Token based F1(%)
Second run						
BD	72.27%	63.74%	67.74%	83.19%	85.02%	84.10%
AC	51.82%	45.71%	48.58%	61.59%	62.94%	62.26%
Third run						
AC	66.85%	64.82%	65.82%	71.63%	71.28%	71.45%

the AC errors. A more completed analysis will be carried out when dataset will be available.

The overall SRL process is, on a standard architecture, performed at about 6.74 sentences per second, i.e. 6.21 sentence per second. For more details we refer the reader to [3] in which same SRL tool is evaluated over the FrameNet (English) dataset.

References

1. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Language Resources and Evaluation http://dx.doi.org/10.1007/s10579-009-9081-4
2. Coppola, B., Moschitti, A., Tonelli, S., Riccardi, G.: Automatic FrameNet-based annotation of conversational speech. In: Proceedings of IEEE-SLT 2008. pp. 73–76. Goa, India (December 2008)
3. Croce, D., Giannone, C., Annesi, P., Basili, R.: Towards open-domain semantic role labeling. In: ACL'10. pp. 237–246 (2010)
4. Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. Computational Linguistics 28(3), 245–288 (2002), http://www.cs.rochester.edu/ gildea/gildea-cl02.pdf
5. Heyer, L., Kruglyak, S., Yooseph, S.: Exploring expression data: Identification and analysis of coexpressed genes. Genome Research pp. 1106–1115 (1999)
6. Johansson, R., Nugues, P.: The effect of syntactic representation on semantic role labeling. In: Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1. pp. 393–400. Proceedings of COLING, Manchester, UK (August 18-22 2008)
7. Landauer, T., Dumais, S.: A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review 104(2), 211–240 (1997)
8. Pado, S.: Cross-Lingual Annotation Projection Models for Role-Semantic Information. Ph.D. thesis, Saarland University (2007)
9. Padó, S., Lapata, M.: Dependency-based construction of semantic space models. Computational Linguistics 33, 161–199 (June 2007), http://dx.doi.org/10.1162/coli.2007.33.2.161
10. Pradhan, S.S., Ward, W., Martin, J.H.: Towards robust semantic role labeling. Computational Linguist. 34(2), 289–310 (2008)