Exploiting the Semantic Fingerprint for Tagging ”Unseen” Words

Fabio Massimo Zanzotto and Armando Stellato

Department of Computer Science
University of *Roma*, *Tor Vergata*, Roma (Italy)
{zanzotto,stellato}@info.uniroma2.it

Abstract

In this paper we want to investigate the use of external and ”orthogonal” semantic resources in building coarse-grained semantic taggers. Our aim is to reduce the degree of supervision for the learning phase by keeping small the set of words whose behaviour has to be manually studied throughout a corpus. We introduce the notion of *semantic fingerprint* in order to exploit these external semantic resources in both machine learning and statistical models. Semantic fingerprints allow a straightforward integration of hierarchical information in the feature vector model. We will study and experimentally compare the effect on coarse-grained semantic taggers of different kinds of semantic fingerprints based on different semantic resources.

1. Introduction

Words seem to be *semantically conservative* as they tend to keep their preferred sense when taken in topically coherent document collections. This intuition underlies many studies in word sense disambiguation as (Madhu and Lytle, 1965; Gale et al., 1992; Resnik, 1997). Let us take, for instance, the famous example of the word *bank*. Even if this word is highly ambiguous, i.e. it has the senses of *institution*, *building*, and *river bank*, a semantic tagger can easily choose the correct sense if the knowledge domain is given. When dealing with texts related to financial news the most probable tag would be *institution*. On the other hand, whenever analysing navy related bulletins it is likely that the word assumes the *river bank* sense. There is some evidence for this phenomenon and it seems to be even very intense when coarse-grained semantic dictionaries are used. In the portion of the British National Corpus tagged with respect to a subset of the LDOCE categories the semantic tagging activity has a perplexity close to 1 (Guthrie et al., 2004).

Exceptions to the word attitude of being semantically conservative seem to be rare. Given the above, the best (and simplest) starting point in building a semantic tagger for a given knowledge domain seems to be collecting a good estimation of the prior distribution of word semantic tags in that specific domain. This estimation would require that, in a new domain, each word is observed and tagged in a sufficient number of instances in order to derive the most likely sense.

In this paper we investigate the possibility of reducing the words over which this manual tagging activity should be done. The manual semantic tagging done for a portion of the dictionary words in the domain corpus should be used to give hints to an automatic classifier in order to discover the most probable semantic tag for the remaining words. For instance, the preferred *investor* sense for the word *bear* in a financial domain (discovered and imposed by manually tagging word instances in the text collection) should help to deduce the same preference for the word *bull*. We claim that, when building a semantic tagger based on a coarse-grained semantic dictionary D, such a kind of beneficial effect may be obtained using a external and more fine-grained lexical resource D'. To investigate this claim we introduce the notion of *semantic fingerprint* as a way to exploit hierarchical semantic information in the classical machine learning feature vectors. After a short discussion on the envisaged procedure for building a semantic tagger (Sec. 2.), we will describe how the semantic fingerprint notion is useful for introducing hierarchical semantic knowledge in the classical feature vector model underlying many machine learning algorithms (Sec. 3.). Then, we will introduce the probabilistic classifiers used to investigate the usability of the semantic fingerprint when building semantic taggers (Sec. 4.). Finally, results of the experimental investigation are discussed (Sec. 5.).

2. Building a semantic tagger for a knowledge domain

The knowledge domain where words are used seems to give relevant hints to infer their sense. In early Machine Translation projects, this information was used to prepare *ad hoc* domain dictionaries containing only word senses relevant for the particular domain (e.g. (Oswald and Lawson, 1953)). Eliciting senses from the dictionary to build a domain sense tagger is not a perfect solution, as domain does not eliminate ambiguity for some words (as noticed in (Dahlgren, 1988)) and as some rare word senses may appear. However, it would be unreasonable not to take into consideration the bias induced by specific domains. For this reason, though all of the word senses have to be kept in our dictionary, domain sense preference for words should be included in a semantic tagger and used to modify sense distribution accordingly. Domain bias may be included in a probabilistic form.

In a closed world assumption, largely done in word sense disambiguation and in semantic tagging (Ide and Veronis, 1998), a dictionary D is used to describe all the necessary word senses. The prior distribution of senses for a word is generally uniform. The exploitation of the domain priming information requires therefore the re-estimation of the sense distribution for each word in the dictionary over the particular domain. As a knowledge domain is often represented as a coherent document collection, the sense distribution has to be estimated observing words in their context. This manual work should be done for each word in the dictionary that is likely to appear in the corpus. This activity will constitute the supervision for the semantic tagging building procedure.

In line with other approaches, our aim is to investigate procedures for building semantic taggers that are open to the reduction of the amount of supervision. Let us examine the general procedure for building a tagger in the closed world assumption. Given a semantic dictionary D with its semantic tag catalogue T and an unnannotated domain corpus C, the target of the procedure is to build the classification function $Tagger(i)$ that assigns the correct semantic tag $t \in T$ to each word instance i for the domain. The building model is the following:

- divide the dictionary D in two halves, namely $Train$ and $ToTag$
- annotate the occurrences of the words belonging to $Train$ in the corpus C
- train a classifier $Tagger$ on the instances of $Train$ in the corpus C
- tag the *unseen* word instances with the trained classifier $Tagger$, i.e. the instances of the $ToTag$ words in the the corpus C

It is worth noticing that the procedure has the ultimate aim to decide the preferred sense for each word (with respect to the semantic catalogue T) in the corpus C, that is representative of the target knowledge domain.

According to the desired degree of unsupervision, the first step of the procedure may be pursued in many ways. As a first possible choice, the dictionary may be randomly divided into two halves. In an active learning environment, the $Train$ section should include the most informative words, e.g. the most frequent words in the corpus C. Finally, in a completely unsupervised approach words in $Train$ may be the unambiguous words in the dictionary D while words in $ToTag$ are all the ambiguous ones. Ambiguity should be defined with respect to the target semantic dictionary. In this latter case the activity of tagging the word instances in the corpus C is eliminated.

In this general procedure, the real problem is to decide what information the classification algorithm $Tagger$ has to rely on. We will try to demonstrate that the use of lexical semantic resource D' other than the dictionary D helps in increasing the performances of the semantic tagger. In this paper we will focus only on information related to the word to be tagged, neglecting all the contextual evidence that could help in the disambiguation process.

Given therefore the external resource D' with its semantic tags T', our basic idea is that words appearing with a given frequency in the corpus shape the behaviour of the other words as some nodes in T' will be more active than the others. If T' is more fine-grained than S or represents an ”orthogonal” semantic model, it should help in classifying words with respect to T (see the above example between *bear* and *bull* in an financial domain).

3. Classification Function and Semantic Fingerprints

What we are seeking is a classification function $Tagger(i) = t$ that proposes a class t for any given instance i representing a word in a text. This classification function will observe objects in an instance space I assigning a class t in a set of possible categories T, i.e.:

$$Tagger : I \rightarrow T$$

In machine learning, this function assumes a variety of shapes, (e.g. decision trees in (Quinlan, 1993)), whereas in a probabilistic framework (e.g. the Maximum Entropy model (Jelinek, 1998)), it is seen as a selector of the most probable category given the conditions imposed by i, i.e.:

$$Tagger(i) = argmax_{t \in T} P(t|i)$$

Obviously, the categorisation is possible if some regularities appear in the space of the instances I. These regularities can be detected whenever observable features are defined. Given the observable features F_1,...,F_n, an instance $i \in I$ identifies a point in the space $F_1 \times ... \times F_n$, i.e.:

$$i = (f_1, ..., f_n) \in F_1 \times ... \times F_n$$

In machine learning this model is generally called feature-value vector and underlies many algorithms, as the ones gathered in (Witten and Frank, 1999).

With this general model in mind, we will try to describe in the rest of the section how an external semantic resource based on an hierarchical organisation can be used. We will firstly concentrate on the general limitations of the feature vector with respect to this problem and we will then propose a possible solution that we call *semantic fingerprint*.

3.1. Features of the feature vector

Many machine learning algorithms (as the ones in (Witten and Frank, 1999)) use the feature-value model assuming:

- the *a-priori independence*: each feature is *a priori* independent from the others and, therefore, no possibility is foreseen to make explicit relations among the features;
- the *flatness* of the set of the values for the features: no hierarchy among the values of the set is taken into consideration;
- the *certainty of the observations*: given an instance I in the feature-value space, only one value is admitted for each feature.

Under these limitations ML algorithms offer the possibility of selecting the most relevant features that may help in deciding whether or not an incoming object in the feature-value space is instance of a given concept.

Exploiting the feature-value vector model and the related learning algorithms in the context of natural language processing may then be a very cumbersome problem, especially when the successful bag-of-word abstraction (Salton and Buckley, 1988) is abandoned for deeper language interpretation models. The a-priori independence among features, the flatness of the values, and the certainty of the observations are not very well suited for syntactical and semantic models. On the one side, syntactical models would require the possibility of defining relations among features in order to represent either constituents or dependencies among words. On the other side, a semantic interpretation of the words (intended as their mapping in an is-a hierarchy such as WordNet (Miller, 1995)) would require the possibility of managing hierarchical value sets in which the substitution of a more specific node with a more general one can be undertaken as a generalisation step. Finally, the ambiguity of the interpretations (either genuine or induced by the interpretation model) stresses the basic assumption of the *certainty of the observations*. Due to ambiguity, a given instance of a concept may be seen in the syntactic or the semantic space as a set of alternative observations. The limits of the underlying interpretation models in selecting the best interpretation requires specific solutions to model *uncertainty* when trying to use feature-value-based machine learning algorithms for learning concepts represented by natural language expressions.

3.2. Hierarchies in the Feature Vector: the Semantic Fingerprint

The use of a hierarchical lexical resource is really cumbersome especially when coupled with the uncertainty of the observations. If we want to rely on an external semantic resource, we surely cannot assume that the activity of reducing the possible senses of the word to one is done before an eventual semantic tagger is in place. Therefore, both *flatness* and *certainty of the observations* represent a problem to be resolved.

Having a lexical hierarchy H associated to the semantic dictionary D', in absence of information the only way is to give a weight to all the active senses (as done in (Resnik, 1997) where a study of word lexical preferences is done). If a word activates n nodes in the hierarchy H each node will cumulate a $1/n$ weight in the classification function whenever encountered as training instance. For the problem we are addressing here, this model seems to disperse too much observations due to the dimension of the feature space that represents all the nodes of the hierarchy H.

We propose to use a subset of the hierarchy that we call *semantic fingerprint* subset. The *semantic fingerprint* of a word should represent all its active senses with respect to this cut of the hierarchy. Then given a hierarchy H underlying a semantic dictionary D' and a subset of nodes SF retained as a useful level of generalisations the *semantic fingerprint* of a word w, i.e. $SF(w)$, is the subset of SF activated by the word w, i.e.:

$$SF(w) = \{s \in SF | s \text{ generalises } s' \text{ and } s' \in senses(w)\}$$

where $senses(w)$ are all the senses activated by the word w in the considered hierarchy H. The set SF represents the semantic tag catalogue of the resource D', i.e. $SF = T'$.

The feature spaces we want to consider should then integrate the word and this semantic fingerprint. Two approaches are possible: a boolean and a weighted activation. The first approach tries to use the semantic fingerprint information and it is a viable solution for many ML algorithms. The second one tries to capture the relative importance between highly unambiguous and polysemous words in the training phase. Given W as the set of all the words of the dictionary and a $S_i = [0, 1]$ real interval for each element s_i in the semantic fingerprint SF, the resulting feature space is:

$$W \times S_1 \times ... \times S_n$$

where n is the cardinality of SF. The boolean model is a subcase of this as it uses only the extremes of each G_i interval. A word w instance i activating a semantic fingerprint $SF(w)$ will then have two possible representations in the feature space. The boolean activation scheme foresees w as first element and 1 for each S_i whose corresponding s_i is in $SF(w)$ and 0 for the others. The weighted activation scheme will have w as first element and $1/|SF(w)|$ for each S_i whose corresponding s_i is in $SF(w)$ and 0 for the others.

One important issue is to understand which is the most relevant semantic fingerprint. This requires to adopt different external lexical resources and different levels of generalisation, i.e. different D' and different SF within the chosen D'.

4. Probabilistic classifiers

We tested the usability of the semantic fingerprint in a probabilistic framework in order to take also profit of the weighted model. As the target is to define the classification function (1), we tried with two different stochastic estimators: a modified maximum likelihood model that takes into account the *uncertainity* of the observations and a maximum entropy model. The sample space over which probabilities have to be estimated is then the following:

$$T \times W \times S_1 \times ... \times S_n$$

where T is the set of all the semantic classes.

For the purpose of the description of the probability estimation, for each class $t \in T$ we define the function:

$$t(i) = \begin{cases} 1 & \text{if } t \text{ is the class of the instance } i \\ 0 & \text{otherwise} \end{cases}$$

and for each $s \in SF$ the function:

$$s(i) = \begin{cases} v & \text{if } v \text{ is the value of the related feature S in } i \\ 0 & \text{otherwise} \end{cases}$$

4.1. Using the Maximum Likelihood estimation in a ”back-off” approach

For this first estimation method, the probabilistic classifier is approximated with:

$$Tagger(i) \approx argmax_{t \in T} \hat{P}(t|i)$$

This latter is estimated as $\hat{P}(t|i) = max_{s \in i} P(t|w, s)$ where w is the word in i while s is one of the generalisation of w in $SF(w)$.

The estimation is then done with the following back-off model that considers the word association with the class

Test Set	*Sem. Fingerprint*	*MaxLik*	*MaxLik weighted*	*MaxEnt weighted*
Light	w	0.7748	0.7748	0.8068
	w + synset	0.7853	0.7866	0.8201
	w + BC	0.8685	0.8698	0.8673
	w + TM	0.8282	0.8527	0.8496
	w + LDOCE	0.8282	0.8201	0.8335
Hard	w	0.6830	0.6830	0.5852
	w + synset	0.6317	0.6114	0.6568
	w + BC	0.7337	0.7371	0.7342
	w + TM	0.6998	0.7002	0.7182
	w + LDOCE	0.6643	0.6608	0.6914

Table 1: Experimental results

more reliable then the generalisations of the word in the semantic fingerprint:

$$P(t|w,s) = \begin{cases} \hat{P}(t|w) & \text{if } w \text{ is a seen word} \\ \hat{P}(t|s) & \text{otherwise} \end{cases}$$

The probabilities are then estimated with the maximum likelihood model as follows. Having a set of training examples Tr, the estimated probability $\hat{P}(t|w)$ is straightforwardly obtainable as:

$$\hat{P}(t|w) = \frac{counts_{Tr}(t,w)}{counts_{Tr}(w)}$$

On the other hand, the probability for the generalisation in the semantic fingerprint is estimated as:

$$\hat{P}(t|s) = \frac{\sum_{i \in Tr} t(i)s(i)}{\sum_{i \in Tr} s(i)}$$

It is worth noticing that the estimators are correctly defined for both the boolean and the weighted scheme.

4.2. Using the Maximum Entropy approach

In the Maximum Entropy model, observable features of instances are called *feature functions*. These are functions that fire in given conditions and allow the detection of some given preconditions (see (Jelinek, 1998)). Given the pair of glasses on the instance space that we have called feature-value vector, an equivalent representation can be found in terms of feature functions. The binary feature function related to the configuation (v, c) has the following form:

$$F\_v\_c(class, i) = \begin{cases} 1 & \text{if } class = c \wedge f_i = v \\ 0 & \text{otherwise} \end{cases}$$

The equivalence between a feature vector and a set of feature functions is thought in terms of representative power. If F is the i-th feature in the feature-value space, in order to represent it we will need $|F| \cdot |C|$ *feature functions* if all the configurations (v, c) with $v \in F$ and $c \in C$ are admissible. It is worth noticing that the set of feature functions can be reduced if some of these configurations are not admissible, i.e. for a given class c the feature F will never assume the value v.

If the space I is observed in the feature-value model, $F_1 \times ... \times F_n$, an equivalent (from the point of view of the expressive power) representation of this model in the ME approach will require $n \cdot |F| \cdot |C|$ feature functions.

5. Experimental Evaluation

These experiments are built to investigate if the semi-supervised approach presented in Sec. 2. is a viable solution for producing semantic taggers and if the notion of semantic fingerprint is somehow useful. Moreover, a second problem is to demonstrate that an external resource is preferable to a self-referring approach. Finally, within the chosen external semantic resource it is necessary to understand which is the more profitable cut of the hierarchy among all the possible ones.

The experiments are carried out using the annotated corpus produced in (Guthrie et al., 2004) where the target is to produce a semantic tagger able to tag with LDOCE categories. In line with what done in (Guthrie et al., 2004), we prepared two different experimental set-ups:

- a *light* test whose words kept apart in the $ToTag$ set are 194 highly ambiguous words
- an *hard* test representing the fully unsupervised model where $Train$ are all the unambiguous words of the dictionary and $ToTag$ are all the ambiguous ones

In the *light* test set the $training$ and $testing$ instances for the classification models have been obtained in the following way: the overall corpus C has been divided randomly in two parts C_1 and C_2. All the instances C_{ToTag} of the words of $ToTag$ in C_1 have been collected. The training instances Tr are then $Tr = C_1 - C_{ToTag}$ while all the testing instances Ts are $Ts = C_2 \cup C_{ToTag}$. On the other hand, in the *hard* test set, $Train$ is the portion of the dictionary that contains the unambiguous words while $ToTag$ is the set of all the ambiguous words. The Tr set is represented by all the instances in C of $Train$ words and Ts gathers all the instances in C of the $ToTag$ words.

The external semantic resource used in the experiments is WordNet and we tried three different semantic fingerprints for the nouns: (1) the synset level, no generalisation is applied and words activate their synsets; (2) the *basic concept* level, a set of WordNet synsets considered in the inter-lingual interface of EuroWordNet (Vossen, 1998); (3) the WordNet topmosts. In Table 1 these semantic fingerprints are respectively called $synset$, BC, and TM.

Two control experiments have been also carried out: one in absence of any semantic fingerprint and the second with a self-referring semantic fingerprint. Table 1 reports the results. It is possible to observe that in the case of the light experiment any use of semantic fingerprint gives a positive gain with respect to the experiment without any generalisation. Moreover, using the generalisation of an external resource is more positive than using a self-referred semantic fingerprint. It is worth noticing that the best semantic fingerprint seems to be based on the EuroWordNet base concepts. The second set of experiments on the hard test provides even more evidence on this relevant observation.

6. Conclusion

In this paper we proposed a way to use an external semantic resource in the process of semantic tagging. This has been integrated in the semantic classifiers using the notion of semantic fingerprint. With the experimental results we demonstrated that use of the semantic fingerprint helps in classifying ”unseen” words, i.e. words whose behaviour has not been manually tagged. The use of an external resource based on a more fine-grained dictionary seems to be a good solution to speed up the production of both general and domain specific semantic taggers.

7. References

Dahlgren, Kathleen G., 1988. *Naive Semantics for Natural Language Understanding*. Boston: Kluwer Academic Publishers.

Gale, William, Kennet Church, and David Yarowsky, 1992. One sense per discourse. In *Proceedings of the Speech and Natural Language Workshop*. San Francisco.

Guthrie, Louise, Roberto Basili, Fabio Massimo Zanzotto, Kalina Bontcheva, Hamish Cunningham, Marco Cammisa, Jerry Cheng-Chieh Liu, Jia Cui, Cassia Farria Martin, David Guthrie, Kristiyan Haralambiev, Martin Holub, Klaus Machery, and Fredrick Jelinek, 2004. Large scale experiments for semantic labeling of noun phrases in raw text. In *Proceedings of the Language, Resources and Evaluation LREC 2004 Conference*. Lisbon, Portugal, fortcoming.

Ide, Nancy and Jean Veronis, 1998. Word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1–35.

Jelinek, Fredrerik, 1998. *Statistical Methods for Speech Recognition*. Cambrige, Massachussetts, USA: The MIT Press Massachusetts Institue of Technology.

Madhu, Swaminathan and Dean Lytle, 1965. A figure of merit technique for the resolution of non-grammatical ambiguity. *Mechanical Translation*, 8(2):9–13.

Miller, George A., 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Oswald, Victor A. Jr. and Richard H. Lawson, 1953. An idioglossary for mechanical translation. *Modern Language Forum*, 38(3/4):1–11.

Quinlan, J.R., 1993. *C4.5: Programs for machine learning*. San Francisco, CA: Morgan Kaufmann.

Resnik, P., 1997. Selectional preference and sense disambiguation. In *Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How?, Washington, April 4-5, 1997.*.

Salton, G. and C. Buckley, 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

Vossen, P., 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Dordrecht: Kluwer Academic Publishers.

Witten, Ian H. and Eibe Frank, 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Chicago, IL: Morgan Kaufmann.