Exploiting the feature vector model for learning linguistic representations of relational concepts

Roberto Basili, Maria Teresa Pazienza, Fabio Massimo Zanzotto
University of Rome ”Tor Vergata”,
Department of Computer Science, Systems and Production,
00133 Roma (Italy)
{basili, pazienza, zanzotto}@info.uniroma2.it

Abstract

In this paper we focus our attention to the construction of one-to-many mappings between the coarse-grained relational concepts and the corresponding linguistic realisations with an eye on the problem of selecting the catalogue of the coarse-grained relational concepts. We here explore the extent and nature of the general semantic knowledge required for the task, and, consequently, the usability of general-purpose resources such as WordNet. We propose an original model, the verb semantic prints, *for exploiting ambiguous semantic information within the feature vector model.*

1 Introduction

Relational concepts and their linguistic realisations are very relevant bits of semantic dictionaries. These equivalence classes, often called *semantic frames*, may enable sophisticated natural language processing applications as argued in [7] among others. For example, take the relational concept *have-revenues(AGENT:X, AMOUNT:Y, TIME:Z)* and two related ”generalised” forms *X has a positive net income of Y in Z* and *X reports revenues of Y for Z*. This would help in finding answers to very specific factoid questions such as *”Which company had a positive net income in the financial year 2001?”* using text fragments as *”Acme Inc. reported revenues of $.9 million for the year ended in December 2001.*”.

Information Extaction (IE) is based on this notion. Templates are relational concepts and extraction patterns are linguistic relatisations of templates or, eventually, of intermediate relational concepts, i.e. the events. Besides used techniques we can say that IE is a *semantic-oriented* application.

Generally, such a kind of applications rely on complete semantic models consisting of: a catalogue of named entity classes (relevant concepts) as *Company*, *Currency*, and *TimePeriod*; a catalogue of (generally) coarse-grained relational concepts with their semantic restrictions, e.g. *have-revenues(AGENT:Company, AMOUNT:Currency, TIME:TimePeriod)*; a set of rules for detecting named entities realised in texts and assigning them to the correct class; and, finally, a catalogue of one-to-many mappings between the coarse-grained relational concepts and the corresponding linguistic realisations. These semantic models are often organised using logical formalisms (as in [6]). The results are very interesting artifacts conceived to represent equivalences among linguistic forms in a systematic and principled manner.

Besides the representational formalism, the actual content of semantic models is a crucial issue. Using *semantic-oriented* systems requires the definition of the relevant semantic classes and their one-to-many mappings with the linguistic realisations within the target knowledge domain. Even if repositories of general knowledge about the world exist both at the concept level (e.g. Wordnet [11]) and at the relational concept level (e.g. Framenet [2]), they can be hardly straightforwardly used. Specific domains and information needs such as airplane travels in [16] or the company mergers and acquisitions in [1] generally stress their limits. Good coverage of phenomena and, consequently, good performances of final applications can be reached when the underlying semantic models are adapted to target domains.

It is reasonable to hope that the cost of building domain-specific semantic resources can be drammatically reduced as such a kind of knowledge already exists in ”natural” repositories: the domain corpora. We are interested in investigating this problem relying on a ”terminologial” perspective [5]. It is our opinion that typical insights of terminology studies as *admissible surface forms* and *domain relevance* help in concentrating the attention on relevant and generalised text fragments when mining large text collections.

In this paper we focus our attention to the construction of one-to-many mappings between the coarse-grained relational concepts and the corresponding linguistic realisations with an eye on the problem of selecting the catalogue of the coarse-grained relational concepts. As it will be clarified in Sec. 2 that describe the terminological approach, we work on a list of extraction patterns derived from the analysis of the domain corpus and we attack an aspect of this twofold problem: the assignment of the correct relational concept given a prototypical linguistic realisation. That is, given a prototypical form *"Company has a positive net income of Currency in TimePeriod"* find *have-revenues* as the correct semantic frame. We leave apart the problem of mapping arguments to thematic roles.

We here explore the extent and nature of the general semantic knowledge truly required for the task, and, consequently, the usability of general-purpose resources such as WordNet [11]. We propose to use well assessed machine learning algorithms based on the feature vector model to study this problem. Limits of the feature vector model when applied to natural language processing tasks are discussed (Sec. 3.1). Trying to overcome these limits we propose an original model, the *verb semantic prints*, for exploiting ambiguous semantic information within the feature vector model (Sec. 3.3). In order to understand the effectiveness of the overall model we study it contrastively with a baseline model based on lexicalised syntatic information (Sec. 3.2). We argue that if general semantic information is relevant we should be able to demonstrate that the related space outperforms the other should across different machine learning algorithms. Moreover, it should demonstrate to better converge to the final classification in unsupervised clustering methods. The experimental investigation is described in Sec. 4. Results over a large range of different machine learning algorithms (collected in [17]) are compared. Finally, before concluding, we briefly discuss the related approaches (Sec. 5) as the problem of finding equivalent linguistic forms for relational concepts is largely debated.

2 A "terminological" perspective in learning equivalent linguistic forms

Domain corpora naturally contain a large quantity of domain knowledge: the same knowledge needed for adapting or building sematic models for semantic-oriented applications. A common practice in terminology extraction [8] is to exploit this knowledge trying to study what emerges from the textual collections. The problem there is to build terminological dictionaries containing relevant concepts, i.e. *terms*.

Our target is to examine domain corpora in order to find relevant *relational concepts* (i.e. semantic frames) and their corresponding linguistic realisations. In analogy with terminological studies, we define a notion of *admissible surface form* (i.e. prototypes for possible textual representations) for relational concepts. Genereally prototypes are given at the synatctic level. We expect that the linguistic forms of *relevant* relational concepts regularly emerge from a possibly complex (but domain independent) corpus analysis process. This can help in both deciding the relevant relational concepts and finding the one-to-many mappings with the linguistic realisations. In this process the following steps are undertaken:

1. *Corpus processing*: the *admissible surface forms* are detected and syntactically normalised. Each normalized form is a generalization of several observations ranked according to their *domain relevance* (i.e. their frequency).

2. *Concept formation*: the most important normalised forms are selected and they provide the set of target conceptual relationships. We will refer to this set as T.

3. *Form classification*: the generalised forms are classified according to the types defined in T.

The notions of *admissible surface forms* and of *domain relevance* used in the corpus processing phase are borrowed from the terminology extraction practice. These are very useful in concentrating the efforts only on relevant analysed text fragments.

As we want to analyse relational concepts we will limit our attention here to verb phrases. Our admissible surface form will be a verb with all his arguments. Even if verb phrases do not cover all the possible relational words, these are very good indicators. We will assume that the concepts, i.e. the catalogue of the named entities and the terms, are given.

The first phase is done more or less automatically using the technique introduced by [5]. Then, domain experts, exposed to the data such as the ones in Tab. 1 sorted according to their relevance (e.g. computed on the frequency $freq$ of the form), can define the relational concepts. In the example *Cooperation/Splitting among Companies* (2-1) or *Market trends* (6-1) can be the two relational concepts formed in this phase. Finally, the classification of the instances is done accordingly, i.e. the column *relational concept* is compiled. The *concept formation* phase is naturally more difficult than the actual classification even if in this phase the concepts as 6-1 and 2-1 are defined extensionally. In the *classification phase* experts using the surrogate forms are able to decide the concept extension on the basis of the observable features such as *percent_ne* (*percentage*), *entity_ne* (*named entity*), *share*, *fall*, *lose*, *join*, or *own*.

freq	*generalised form*	*relational concept*
88	(subj,entity_ne) own (dirobj,percent_ne)	1-2
70	(subj,entity_ne) join (dirobj,entity_ne)	1-2
58	(subj,entity_ne) lose (dirobj,percent_ne)	6-1
47	(subj,share) fall (dirobj,percent_ne)	6-1

Table 1. A very small sample of the classified admissible forms

3 Syntactic feature space and verb semantic prints for learning relational concepts

The purpose of this study is trying to imitate experts in forming relational concepts and in classifying linguistic forms using well-assessed machine learning algorithms. We want also to investigate the role in the task of general semantic knowledge (i.e. Wordnet). Before developing new algorithms for a task it can be useful to understand if the feature observation space is worthy. However, the basic problem that arises when using existing machine learning algorithms is to understand if the underlying model, i.e. the feature-value vector and its usage, supports the observations we want to model. Before describing the syntactic (Sec. 3.2) and the sematic model (Sec. 3.3) we propose for form classification and, eventually, for relational concept formation, we examine the limitations of the feature-value vector model when used over models for natural language (Sec. 3.1).

3.1 Feature-Value Vector vs. Syntax and Concept Hierarchies

A largely used model for describing instance characteristics is the feature-value vector. This model underlies many machine learning algorithms as the ones gathered in [17]. It suggests an observation space in which dimensions represent features of the object we want to classify and dimension values are the values of the features as observed in the object. Each instance object is then a point in the feature space, i.e. if the feature space is $(F_1, ..., F_n)$ an instance I is:

$$I = (f_1, ..., f_n) \tag{1}$$

where each f_i is respectively the value of the feature F_i for I.

Many machine learning algorithms (as the ones in [17]) 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 in 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 they offer the possibility of selecting the most relevant features that may decide 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 processing natural language may then be a very cumbersome problem especially when the successful bag-of-word abstraction [15] 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 [11]) 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 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 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 A very simple syntactic (lexicalised) model

As we have seen in Sec. 2, the objects to be classified are generalised verb forms, i.e. verbs with their more frequent arguments. Apparently, it can seem very simple mapping those structures to the feature-value vector. The verb and the more stable arguments are in fact highlighted and, moreover, the arguments are classified according to the played syntactic role. A straightforward mapping can therefore be performed and this is what we did. We call this space as the syntactic-lexicalised feature space, hereafter referred as *synt-lex* space. The selected features (for the feature-value vector model) are then respectively the verb, the subject, the object, and finally the remaining arguments represented by their heading preposition. This defines the feature vector $(F_1, ..., F_n)$. Each pattern prototype $(v, \{(arg_1, lex_1), ..., (arg_n, lex_n)\})$ has therefore a mapping to a feature-value vector in the following way. Each F_i has the value:

$$f_i = \begin{cases} v & \text{if } F_i = verb \\ lex_j & \text{if } \exists j.F_i = arg_j \\ none & \text{otherwise} \end{cases} \tag{2}$$

It is worth noticing that in the case of the prototype forms the syntactic ambiguity is not a problem. These patterns are in fact abstractions of the behaviour of the verbs in the corpus, i.e. its arguments are statistically filtered. Furthermore, in the corpus processing phase, first of all stable generalised noun phrases are detected. This helps to filter out possibly frequent wrong verb attachments detected by the syntactic parser. Therefore, each item in the verb prototype form is then unambiguously considered as verb argument.

The chosen mapping method has some inherent limitations. Firstly, the structure of the complex noun phrases is not resolved in the feature-value model. They are in fact preserved as they are, i.e. the overall structure is replicated in the value of the related feature. For instance, in the case of *(subject,share_of_companyNE)* where a complex noun phrase appear the value given to the $subject$ feature is exactly the related form. The main reason for this choice is that the more complex structure is more selective for classifying incoming instances. However, no subsumption is possible between the form *share_of_companyNE* and *share*. Such instances will be considered as completely different forms. The second limitation is instead introduced by the ”variable drop” we perform in building the verb pattern prototypes. As part of the semantic of the verb is given by its surface syntactic structure [10], we tend also to offer relevant partially incomplete verb pattern prototypes where the lexicalisation of some syntactic argument may be left ungrounded. The annotators may face a pattern as the following:

(fall,{(subject,ANY),(from,currencyNE),(to,ANY))

where some arguments of the verb are indicated but no restriction is given (i.e. ANY lexicalisation or named entity class is admitted). Possibly using the expectations induced by the investigated domain, the annotator should decide whether or not the given information helps in classifying the instance. In some case, a decision may be also taken with this reduced information. However, as no subsumption is possible in the feature-value translation of the instance no explicit relation may be drawn with an other instance such as (fall,{(subject,share)}). This sort of variable drop cannot be managed. Finally, the mapping solution we adopted does not take into account the possible syntactic changing of the arguments as considered in the method exploited in [9] for a verb paraphrasing algorithm. It is worth noticing that in the case of [9] the search space was reduced by the fact that only couples of verbs suggested by a dictionary have been considered.

3.3 Ambiguous conceptual generalisations as verb fingerprints

The exploitation of conceptual hierarchy is instead a more cumbersome problem due to the limits of the feature-value model. The idea here is to investigate the possibility of integrating some sort of ”semantic” generalisation for the verbs. These latter semantically govern the verbal phrases taken as forms admissible for the relationships and may give an important input to cluster prototype forms in classes. For instance, let us take the patterns in Tab. 1 and suppose that the first three lines have already been encountered, i.e. these can be considered training examples. According to the syntactic-lexicalised space previously defined the new instance may belong both to class 6-1 and to the class 1-2 as it has one common feature with all the considered known instances. The only possibility of classifying the new instance in one of the two classes relies on some sort of generalisation and the verb seems to be a very good candidate. According to WordNet *lose* and *fall* have two common ancestors *change* and *move-displace*. This does not happen for *fall* and *join* or *fall* and *own*. The injection of such a kind of knowledge seems therefore to be useful for the classification task as happens in [4, 9] where noun conceptual hierarchies have been exploited using the definition of distance measures among nodes.

The introduction of a conceptual hierarchy is somehow in contrast with what has been above called the *flatness* of the feature values. If we want to use this information, this hierarchies should be somehow reduced to a flat set where the problem of the inherent structure is simply forgot. One possibility is choosing one level of generalisation and reducing each element to this level. This is the one we adopt in our model for the exploitation of conceptual hierarchies in the problem of detecting equivalent surface forms. In particular, in order to limit the number of features we have chosen the level of the topmosts, hereafter referred as the set T.

If the previous choice helps in using part of the hierarchy, there is still the issue of the ambiguity that in this case cannot be neglected. We do not plan to use any a priori word sense disambiguation mechanism. We would rather prefer to discover and limit the senses of the investigated verb a posteriori, i.e. while analysing the verb prototype forms. Verb senses should be determined in the domain defined by the text collection. The ambiguity should then be modelled in the images of the pattern prototypes in the feature space. It is as if we model uncertainty in the observations of concept instances. However, features can not have multiple values. The way we propose in our model to solve the problem is to use all the topmost senses activated by the analysed verb as representing of the ”overall sense” of the verb. This set can be considered as *verb semantic print*. It will be the task of the machine learning algorithm the selection of the sense (or the senses) more promising for representing the investigated relationship. The algorithm will therefore also work as verb sense disambiguator if the semantic information and the way we use it demonstrates to be useful.

The second model we propose integrates then syntactic with semantic information. The syntactic semantic space is $(F_1, ..., F_n, T_1, ..., T_k)$ where F_i features and the related f_i values have been defined in the previous section whilst the T_j represent the *verb semantic print*. In particular, all the elements in the topmost set T are represented in the feature space. Given a verb prototype form headed by the verb v, the value $t_i \in \{yes, no\}$ for the each semantic feature T_i in the respective point in the feature space is obtained as follows:

$$t_i = \begin{cases} yes & \text{if } hyper(v, T_i) \\ no & \text{otherwise} \end{cases} \tag{3}$$

where $hyper(x, y)$ is the property defining the hyperonym relation among x and y. This latter space will hereafter referred as syntactic-lexicalised-semantic space(*synt-lex-sem*).

4 Experimental investigation

In the previous sections, we proposed a model for exploiting syntax information and semantic networks in machine learning algorithms. As discussed, the proposed models (and the related feature spaces) relies on a large number of approximations to overcome the limitations of the feature-value model. In this section, we will explore the performances the machine learning algorithms will obtain relying on the proposed models in order to understand the relevance of the syntactic and semantic information. First of all, we will describe the test set preparation. This will clarify the final classification task. Secondly, the performances of a number of machine learning algorithms will be analysed over the two proposed feature-value space, i.e. *synt-lex* and *synt-lex-sem*. In this latter phase we will use well-assessed machine learning algorithms gathered in Weka [17]. This collection of algorithms, originally done for Data Mining, has the principal advantage of proposing stable input interfaces for a large number of algorithms. This speeds up the possibility of testing a large number of different algorithms for the same problem. The cross-algorithm validation can give hints on the relevance of the chosen features and on the correctness of the proposed model.

4.1 Corpus analysis and test-set preparation

As discussed in Sec. 2, the context of the experiment is an overall methodology intended to extract equivalent forms out from a homogeneous document collection, i.e. the domain corpora. It worth noticing that the homogeneity hypothesis seems to be similar to the one driving the methods in [18, 14]. The main difference is the grain: the cited two methods in fact that it is stated for each document the belonging to a very specific class representing the specific information need, conservatively here we are thinking to documents related to a coarse grain class such as *sport*, *finance*, etc. Efficient methods to obtain such a document classification may be settled on the bag-of-word document model [15]. Moreover, such kinds of classified corpora are largely available: news agencies and on-line newspapers tend to offer documents organised in a classification scheme to better serve their costumers.

For the reported experiment, we used a corpus consisting of financial news. The text collection gathers around 12,000 news items published from the Financial Times in the period Oct./Dec. 2000. The relational concepts we will discover are therefore the ones related to financial events. After the *corpus processing phase*, that selected around 44,000 forms appearing more that 5 times in the corpus collection, in the *concept formation phase* 13 target relational concepts have been defined inspecting the top ranked forms (see Tab. 2) . Even if we don’t claim this as an exhaustive list, the defined relational concepts represent the more relevant knowledge appearing in the document collection and, more in general, in financial news.

The classification of the forms in the classes has be performed by 2 human experts. Out of the first 2,000 forms considered, 497 were retained as useful, i.e. the information carried in the words or in the named entity classes survived in the form has been considered sufficient to draw a conclusion on the classification. Due to the nature of the overall list of pattern prototypes, some of the more specific forms may be trivially tagged using an eventually classified more general form. In the preparation of the final test set we therefore got rid of this simple cases. When the class of the more specific form it is the same of the more general one, the more specific form has been removed. The resulting test set consists then of 167 different forms whose classification cannot be trivially obtained. The distribution of the forms

		Class	# of equivalent linguistic forms
1		RELATIONSHIPS AMONGS COMPANIES	
	1-1	Acquisition/Selling	15
	1-2	Cooperation/Splitting	8
2		INDUSTRIAL ACTIVITIES	
	2-1	Funding/Capital	4
	2-2	Company Assets (Financial Performances , Balances, Sheet Analysis)	20
	2-3	Market Strategies and plans	
	2-4	Staff Movement (e.g. Management Succession)	6
	2-5	External Communications	13
3		GOVERNMENT ACTIVITIES	3
	3-1	Tax Reduction/Increase	
	3-2	Anti-Trust Control	
4		JOB MARKET - MASS EMPLOYMENT/UNEMPLOYMENT	3
5		COMPANY POSITIONING	
	5-1	Position vs Competitors	3
	5-2	Market Sector	7
	5-3	Market Strategies and plans	7
6		STOCK MARKET	
	6-1	Share Trends	62
	6-2	Currency Trends	0

Table 2. The event class hierarchy of the financial domain and form distribution

in the classes is reported in Tab. 2.

It is worth noticing that in the final list only 4 macroscopic parsing errors survive: 3 related to prepositional phrase headed by *of* erroneously considered attached to the verb and one related to the form:

(value,{(dirobj,company_at_currencyNE)})

The verb modifier *at_currencyNE* has been erroneously considered as modifier of the noun *company*. This is mainly because the overall form appears frequently as it is and, therefore, the fact that is chosen the ”noun” reading is because this attaching phase is run as the first. These errors have been left in the final list in order to see the robustness of the learning algorithms with respect to spurious input data.

4.2 Analysis of the results

The classification problem over the proposed spaces has been therefore studied with a number algorithms and the results have been reported in tab. 3. It appears that the baseline of the classification problem proposed is around 37% that is reached by those algorithms classifying all the instances in the more probable class (i.e. 6-1). This value of performance is obtained by the NaiveBayes classifier and the DecisionStump. An important observation is that all the other algorithms report even in the syntactic space better results with respect to the base-line, i.e. they are not confused by the provided features. Furthermore, the use of the semantic information by means of the *verb semantic print* seems to be relevant. The major part of the investigated algorithms has an advantage in semantic space. The confusion introduced by the ambiguity seems to be easily managed and the relevant information used. The algorithms are doing the job of disambiguating the verb senses. The best result is obtained by the Voting Feature Interval algorithm on the semantic space. However, it does not seems to have a relevant improvement with the introduction of the semantic. It is worth noticing that this model is statistically based and, when it faces nominal attributes[1] as the one proposed here, it becomes very similar to a profiled based classifier. Looking in the tab. 3, it furthermore seems that algorithms classifying with probability scores (as the NaiveBaye, HyperPipes, and VFI) take a small benefice from using the semantic information as it has been modelled.

Algorithms based on the decision trees (i.e. j48) give moreover the possibility to understand which are the more important attributes driving the decisions. Observing the decision tree for the *synt-lex-sem* space, it becomes clear that the more selective information is represented by the verb senses. Verb lemmas nearly disappeared, i.e. verb senses generalised this information. This phenomenon is not obvious due to the previous independence among the attributes. Furthermore, interesting classification rules as the followings may be observed:

$$
\left(\begin{array}{c}\text{verb}\\\hline change\\ \neg control\end{array}\right)
\left\{\begin{array}{cc}
\left(\begin{array}{c}\text{obj}\\\hline percentNE\\ currencyNE\end{array}\right) & \textbf{6-1}\\
\left(\begin{array}{c}\text{obj}\\\hline capital\\ fund\end{array}\right) & \textbf{2-1}\\
\left(\begin{array}{c}\text{subj}\\\hline turnover\\ income\\ operating\_profit\\ pre-tax\_profit\\ revenue\\ profit\end{array}\right) & \textbf{2-2}
\end{array}\right. \tag{4}
$$

A verb of *change* (but having any sense of *control*) assumes very different meaning according to the companions. This clustering can be a very interesting starting point to write more complex semantic restrictions that tend to cluster also

[1] Attributes assuming values in a finite set.

Method	*synt-lex*	*synt-lex-sem*	% increase/decrease
j48.J48	60.355%	65.0888%	+7,84%
j48.PART	53.8462%	56.8047%	+5,49%
DecisionStump	36.6864%	42.0118%	+14,52%
DecisionTable	59.1716%	59.1716%	0
IB1	47.3373%	60.9467%	+28,75%
IBk	55.6213%	60.9467%	+9,57%
ID3	44.9704%	44.9704%	0
NaiveBayes	36.6864%	37.2781%	+1,61%
HyperPipes	63.9053%	62.7219%	-1,85%
VFI	65.6805%	66.2722%	+0,90%

Table 3. Success rate of different methods over the two spaces in a 5-fold cross-validation

nouns as done in [4, 9].

There is a last consideration in favour of the semantic space. It seems to offer a better possibility of learning this classes from scratch using a clustering algorithm. In the case a very simple algorithm, i.e. the simple k-means with 20 clusters and averaged on 10 different seeds we obtained an error rate of 72.54% for the synt-lex space and 69.17% for the synt-lex-sem space. This timid result induces to think that, in the concept formation phase, better results can be obtained using some sort of semantic model.

5 Related work

It is largely agreed that availability of explicit many-to-one mappings between linguistic forms and their corresponding meaning (i.e. concepts or relational concepts) is beneficial for several linguistic applications. Many researches are in fact devoted to propose methods for automatically building equivalence classes of patterns in fields such as Information Extraction [18, 14], Question Answering [13], Terminology Structuring [12], or Paraphrasing [3, 9].

The automatic construction of equivalent linguistic patterns has been studied attacked from extremely different perspectives and for apparently different reasons. The target relationships range from the very general *hyperonym* relation investigated in automatic approaches to terminology structuring (e.g. [12]) to more specific information as those expressed by equivalence classes of paraphrases [3, 8, 9]. Clearly, template acquisition as typically employed in Information Extraction (e.g. [14]) is part of these studies. The target relationships may vary slightly but the common underlying targets of these methods are equivalence relations derived by analysing text material. The aim is to derive different surface forms of prototypical relationships by means of the smallest annotation effort possible.

In [14, 18] the problem of building information extraction patterns from scarcely annotated texts is investigated. In this case, the target relationship is very complex (i.e. a template) and very specific. Due to the fact that the template is *a priori* known, the notion of *relevance* of the texts in its respect can be suitably exploited. Similarities among the different but *relevant* texts suggest equivalent linguistic forms. The issue of classifying texts is central in the two approaches: in [14] the full classification of the texts in relevant vs. irrelevant is required whilst in [18] a bootstrapping approach is used[2]. It is to be noticed that both methods strongly rely on the shortness of the investigated texts, each one usually targeted to only one template.

A completely different approach to pattern clustering is proposed in [12, 13]. The targets are binary relationships among concepts and the assumption is that (at least some instances of) the related concepts are known *a priori*. When such coupled concepts jointly appear in a text fragment, this latter is assumed as a valid form for the target relationship.

In [12], the method has been used to compile equivalent forms for the *is-a* relationship in the context of terminology structuring. As in any terminology extraction approach the corpus used specifically models a knowledge domain. In [13], the corpus considered has been the entire world wide web and the target was to find the answering patterns using question-answer couples. Questions are first of all (manually) clustered to identify the target relationship types, called here question types (e.g. *inventor*, *discoverer*, etc.). Then for each question the couple *answer* and main *name* of the question are extracted. These latter are used to query an information retrieval engine in order to find the forms representing the given relationships.

In [3] the target is to learn syntactic paraphrasing rules mainly for verbal sentences instead of nominal forms (e.g. as in [8]). The problem is then slightly different but an interesting method for deriving the equivalence among the surface forms is used. In fact, ”parallel corpora”, as those employed in machine translation studies, are collected by

[2] The relevance of texts with respect to a template is modelled as a sort of distance between new texts and a kernel of annotated texts

groupings different English translations of a single non-English text (e.g. a novel). The different translator styles offer heterogeneous translations of the same sentences that in fact convey the same meaning . Parallel sentences thus embody equivalent forms of the same relationship. Although this method is very interesting for general syntactic paraphrasing rules, it has a limited applicability due to the specific ”parallel corpora” employed.

For all the methods, the use of some previous specific knowledge (not always available) seems indispensable:

- focused and structured templates plus examples in [18, 14]
- definitions and examples of the target relationships in [12, 13]
- parallel corpora for [3]

6 Conclusions

In this paper, after the analysis of the limits of the feature-value model, we proposed a method for exploiting well-assessed machine learning algorithm for the problem of learning equivalent surface forms. We obtained some indications that the proposed way to use semantic hierarchies may helpful in the proposed problem. In any case, the overall approach may be included as a suggesting mechanism for the experts involved in the task.

References

[1] D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. Fastus: a finite-state processor for information extraction from real-world text. In *13th International Joint Conference on Artificial Intelligence*, Chambry, France, 1993.

[2] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *Proceedings of the COLING-ACL*, Montreal, Canada, 1998.

[3] R. Barzilay and K. McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th ACL Meeting*, Toulouse, France, 2001.

[4] R. Basili, M. T. Pazienza, and M. Vindigni. Corpus-driven learning of event recognition rules. In *Proceedings of Workshop on Machine Learning for Information Extraction, held in conjunction with the 14th European Conference on Artificial Intelligence (ECAI)*, Berlin, Germany, 2000.

[5] R. Basili, M. T. Pazienza, and F. M. Zanzotto. Learning IE patterns: a terminology extraction perspective. In *Proc. of the Workshop of Event Modelling for Multilingual Document Linking at LREC 2002*, Canary Islands (Spain), 2002.

[6] R. Gaizauskas and K. Humphreys. Using a semantic network for information extraction. *Natural Language Engineering*, 3, Parts 2 & 3:147–169, 1997.

[7] D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, Nov. 2002.

[8] C. Jacquemin. *Spotting and Discovering Terms through Natural Language Processing*. Massachusetts Institue of Technology, Cambrige, Massachussetts, USA, 2001.

[9] N. Kaji, D. Kawahara, S. Kurohashi, and S. Sato. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th ACL Meeting*, Philadelphia, Pennsilvania, 2002.

[10] B. Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, IL, 1993.

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

[12] E. Morin. *Extraction de liens sémantiques entre termes à partir de corpus de textes techniques*. PhD thesis, Univesité de Nantes, Faculté des Sciences et de Techniques, 1999.

[13] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th ACL Meeting*, Philadelphia, Pennsilvania, 2002.

[14] E. Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, Oregon, 1996.

[15] G. Salton. *Automatic text processing: the transformation, analysis and retrieval of information by computer*. Addison-Wesley, 1989.

[16] D. Stallard. Talk’n’travel: A conversational system for air travel planning. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP’00)*, Seattle, Washington, 2000.

[17] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, Chicago, IL, 1999.

[18] R. Yangarber. *Scenario Customization for Information Extraction*. PhD thesis, Courant Institute of Mathematical Sciences, New York University, 2001.