Knowledge Acquisition through Error-Mining

Milagros Fernández Gavilanes Eric Villemonte de la Clergerie
Manuel Vilares Ferro
Computer Science Department, University of Vigo
Campus As Lagoas s/n, 32004 Ourense, Spain
{mfgavilanes,vilares}@uvigo.es
Institut National de Recherche en Informatique et en Automatique
Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France
Eric.De_La_Clergerie@inria.fr

Abstract

We describe an approach to acquiring a knowledge representation applied on technical documents. We focus on corpus with a strong underlying structure, which allows us to follow a number of precise patterns of presentation. Our goal is to provide effectiveness by reducing both time and cost, as well as subjectivity.

Keywords

Knowledge acquisition, parsing, term extraction.

1 Introduction

A number of proposals exploit parsing in order to permit semantic relations to emerge from text, by combining term extraction and term clustering facilities. The former acquire term candidates from tagged corpora through a shallow grammar. Term clustering groups and classifies these candidates in a graph reflecting the relations between them. So, some authors propose conflating candidates that are variants of each other through a self-indexing procedure [7], while others [5] post-process parse trees so as to emphasize the dependency relationships between the content words.

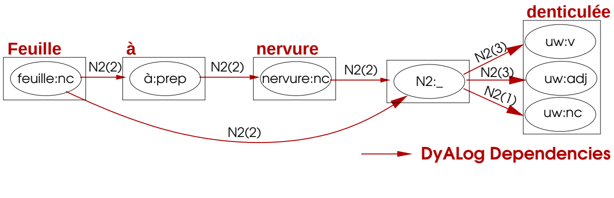

Fig. 1: *Parsing shared-forest from DyALog*

In our approach, the acquisition phase is performed from a *tree-adjoining grammar* (TAG) [8], generated from a source *meta-grammar* (MG) [4]. The clustering phase is performed on the basis of an iterative algorithm inspired by an error-mining strategy [10].

2 The running corpus

We introduce the strategy from a botanic corpus. We concentrate on the *"Flore du Cameroun"*, which is composed of about forty volumes in French, each one running to about 300 pages, organized as a sequence of sections, each one dedicated to one species and following a systematic structural schema. Sections include a descriptive part enumerating morphological aspects such as color, texture, size or form. This implies the presence of nominal phrases, adjectives and also adverbs to express frequency and intensity, and named entities to denote dimensions.

3 The parsing frame

We choose to work with TAGs [8], a grammatical formalism that has given rise to a lot of interest in the modeling of syntax in *natural language processing* (NLP) by combining properties such as the principle of extended domain of locality[1] and a polynomial time complexity, making it appropriate for practical purposes. Using DyALog [3] as parsing frame, we apply a tabular interpretation [1], which implies an efficient treatment of non-deterministic entries.

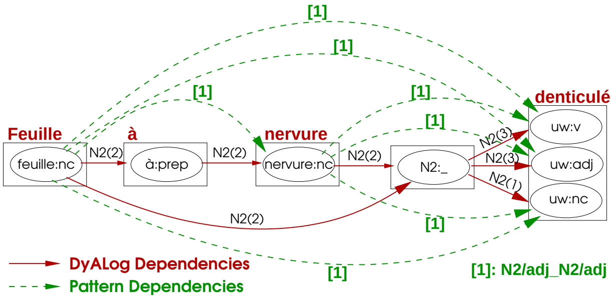

Fig. 2: *Graph of dependencies from DyALog*

The text is parsed on the basis of the MG concept [4], which permits the introduction of a high degree of abstraction in the design of NLP parsers by involving elementary constraints re-grouped in classes, these themselves inserted in a hierarchy of multiple heritage. This allows descriptions to be progressively refined, which is of particular interest when we are describing complex linguistic behavior. DyALog [3] returns

[1] it allows constraints to be defined at more than one level of the parse as compared to context-free rules and permits the use of atomic features.

total or partial parsing shared-forests from a possibly non-deterministic input on the basis of a TAG of large coverage for French, as we can see in Fig. 1 for the sentence *"feuille à nervure denticulée"*, in future our running example. Arrows represent binary dependencies between words through some syntactic construction.

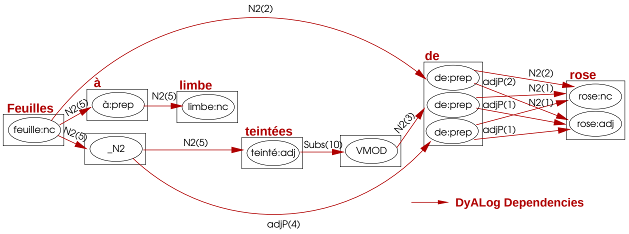

Fig. 3: *Another parsing shared-forest from DyALog*

From this shared-forest, we can extract a graph of dependencies of the type *governor/governed*, as is shown in Fig. 2 by using dotted-lines going from the governor term to the governed one. The probability of a dependency occurence is labeled `P(word1:c1,[label],word2:c2)`, being `word1` the governor word, `c1` the lexical category of the `word1`, `label` the tag of the dependence, `word2` the governed word and finally, `c2` the lexical category of the `word2`. Rectangular shapes represent *clusters*, that is, forms that refer to a position in the input string and all the possible lemmas with their corresponding lexical categories. We call the latter *nodes*, represented by ellipses. Lexical ambiguities correspond to clusters containing nodes with different lemmas, or the same lemma associated to different lexical categories.

3.1 Lexical ambiguities

The morpho-syntactic phase consists of a pipeline named Sxpipe [9], that concatenates a number of tasks such as chunking, entity recognition and tagging.

In spite of the strategy considered, tagging often becomes a non-deterministic and even incomplete task, especially in dealing with an encyclopedic corpus with a high degree of unknown words, as is shown in Fig. 1, where the word *"denticulée"* ("dentate") is initially labeled as unknown word (uw) with three possible associated lexical categories: verb (v), adjective (adj) and common noun (nc). These ambiguities cannot always be solved at lexical level and, in order to avoid prematurely discarding useful interpretations, all the available information should be translated to be considered at parsing time, which introduces an additional factor of syntactic ambiguity.

It is the case of *"feuilles à limbe teintées de rose"* that we could interpret as "rose's tinted laminar leaves", as "rose-tinted laminar leaves" or as "tinted laminar rose leaves". In the first case, the word *"rose"* would be a noun related to *"teintées* ("tinted"), while in the other ones it is an adjective related to *"feuilles"* ("leaves"); as is shown in Fig. 3.

3.2 Syntactic ambiguities

Parsing in NLP is also an incomplete task because it deals with shallow/partial strategies focused on identifying dependencies between terms that are close in the text, as in the case of noun sentences involving:

1. Prepositional attachments, as in *"feuille à nervure denticulée"*, that we could locally translate in two ways: "leaf with dentate vein" or "dentate leaf with vein". It becomes here impossible to establish if the word *"denticulée"* ("dentate") relates to *"feuille"* ("leaf") or to *"nervure"* ("vein"), as is shown in Fig. 1.

2. Coordination structures relating properties to a list of nouns, as [9] in *"des sépales ovales-aigus, glabres ou éparsement hérissés"* ("Sepals oval-pointed, smooth or scattered bristly"), where the property *"hérissés"* ("bristly") could be attached to *"glabres"* ("smooth") or to *"éparsement"* ("scattered").

both of them causing local non-determinism.

4 Knowledge acquisition

Once we recover the graph of dependencies, we extract the latent semantics in the document by compiling additional information from the corpus in order to eliminate useless dependencies. So, the lexical ambiguity in Fig. 3 should be decided in favor of the first alternative ("rose's tinted laminar leaves"), because we have the certainty that plants with rose colored leaves do not exist. Given that we are dealing with a corpus on botany, we should confirm that extreme by exploring it in-depth. That is, to solve the ambiguity we just need the information we are looking for; which leads us to consider an iterative learning process to attain our goal.

In similar terms we describe the syntactic disambiguation process for the example in Fig. 1, by selecting "dentate leaf with vein" as the correct interpretation. Also, we should associate *"hérissés"* ("bristly") to *"éparsement"* ("scattered") in the sentence *"des sépales ovales-aigus, glabres ou éparsement hérissés"* ("Sepals oval-pointed, smooth or scattered bristly"). So, term extraction is the starting point to formalize such a task.

4.1 Term extraction

We consider two principles. Firstly, the *distributional semantic* model [6] establishing that words whose meaning is close often appear in similar syntactic contexts. Also, we assume that terms shared by these contexts are usually nouns and adjectives [2], which means we have chosen to work with a nominal regime.

Term extraction is organized around the recognition of generic lexical and/or syntactic patterns. On the lexical side, we take advantage of linguistic marking information, focusing on conjunctions *"X et X"* ("X and Y"), interval definitions of type *"de X à Y"* ("from X to Y"); or relations involving more explicit physical information such as *"en forme de X"* ("in form of X") or *"de couleur X"* ("of color X"). The result serves to acquire simple concepts such as the value for `color`, `form` or `domain` properties; or to detect enumerations that can propagate some of these values.

1.	$P(\text{feuille:uc, [à-1], nervure:uc})_{\text{local(0)}} = \frac{P_c(\text{feuille:uc})_{\text{local}}\ P_c(\text{nervure:uc})_{\text{local}}\ P_{\text{dep ini}}(\text{feuille:uc, [à-1], nervure:uc})}{\Sigma_{X,Y}\, P_c(\text{feuille:X})_{\text{local}}\ P_c(\text{nervure:Y})_{\text{local}}}$
2.	$P(\text{feuille:uc, [à-1], nervure:uc})_{\text{global(n+1)}} = \frac{\Sigma_{i=1}^{n} P(\text{feuille:uc,[à-1],nervure:uc})_{\text{local(i)}}}{\#\text{dep}_{\text{local(n)}}}$
3.	$P(\text{feuille:uc, [à-1], nervure:uc})_{\text{local(n+1)}} = \frac{P(\text{feuille:uc, [à-1], nervure:uc})_{\text{local(n)}}\ P(\text{feuille:uc, [à-1], nervure:uc})_{\text{global(n+1)}}}{\Sigma_{X,Y}\ P(\text{feuille:X, [à-1], nervure:Y})_{\text{local(n)}}\ P(\text{feuille:X, [à-1], nervure:Y})_{\text{global(n+1)}}}$

Table 1: *Extraction of dependencies for* "feuille à nervure denticulée"

Syntactic patterns revolve around the following relations involving nouns and/or adjectives:

- Noun adjective: like *"feuilles elliptiques"* ("elliptical leaves").
- Noun sth noun: like *"fleur avec pétale"* ("flower with petal").
- Noun sth adjective: like *"pétale avec du rouge"* ("petal with red").
- Adjective adjective: like *"ovale elliptique"* ("elliptical oval").
- Adjective sth adjective: like *"rugueux ou poilu"* ("coarse or hairy").

while other ones, especially involving adverbs, will be considered as future work. So, the vocabulary is concentrated around these terms that from now on we call *pivot terms*.

4.2 Term clustering

We simplify the graph of dependencies in order to obtain the most pertinent ones. We look for these, which we baptize as *strong dependencies*, around pivot terms.

4.2.1 A simple syntactic constraint

We require a simple syntactic constraint establishing that a governed word can only have one governor. So, for example, in the sentence of Fig. 1, *"denticulée"* ("dentate") is governed by *"feuille"* ("leaf"), but also by *"nervure"* ("vein") and, in consequence, we should eliminate one of these dependencies. No other topological restrictions are considered. So, a governor word can have more than one governed one; as in the second interpretation of Fig. 1 ("dentate leaf with vein"), where *"feuille"* ("leaf") is the governor for *"nervure"* ("vein") and *"denticulée"* ("dentate"). Also, one word could be governor and governed at the same time, as is the case of *"nervure"* ("vein"), that is the governor for *"denticulée"* ("dentate"), but is also governed by *"feuille"* ("leaf").

Given that our graph of dependencies is a parse shared-forest, we have chosen to work with a term clustering technique that is inspired by an error-mining proposal originally designed to identify missing and erroneous information in parsing systems [10]. Intuitively, we focus on detecting and later eliminating those dependencies that are found to be less probable in sentences including terms with a low frequency.

4.2.2 The iterative process

We combine two complementary iterative processes. For a given iteration, the first one computes the probability of each dependency; taking as starting point the statistical data provided by the original error-mining strategy and related to the lexical category of the pivot terms. The second process computes, from the former one, the most probable semantic class to be assigned to terms involved in the dependency. So, in each iteration, we look for both semantic and syntactic disambiguation, each one profiting from the other. A fixed point assures the convergence of the strategy [10].

We illustrate term clustering on our running example in Fig. 2, focusing on the dependency labeled [à-1] relating *"feuille"* ("leaf") and *"nervure"* ("vein"); talking without distinction about weight, probability or preference to refer the same statistical concept. So, from Table 1, we have that:

1. To begin with, we compute the local probability of the dependency in each sentence, which depends on the weight of each word, this in turn depending on the word having the correct lexical category. To start the process, first category assumptions, denoted by $\mathtt{P_c}$, are provided by the error-mining algorithm [10]. We take also into account the initial probability for the dependency considered, $\mathtt{P_{dep\ ini}}$, a simple ratio on all possible derivations involving the lexical categories concerned. The normalization is given by the preferences for the possible lexical categories involving each one of the terms considered and here represented by variables X and Y.
2. We re-introduce the local probabilities into the whole corpus locally in the sentences, in order to re-compute the weights of all possible dependencies, estimating then globally the most probable ones. The normalization is given by the number of dependencies connecting the terms considered, #dep.
3. The local value in the new iteration should take into account both the global preferences and the

$$\text{4.}\quad P(\text{feuille:uc:org}, [\text{à-1}], \text{nervure:uc:org})_{\text{local(0)}} = \frac{P(\text{feuille:uc}, [\text{à-1}], \text{nervure:uc})_{\text{local(0)}}\; P(\text{feuille:uc:org})_{\text{local(0)}}\; P(\text{nervure:uc:org})_{\text{local(0)}}}{\Sigma_{X,Y}\, P(\text{feuille:uc:X})_{\text{local(0)}}\; P(\text{nervure:uc:Y})_{\text{local(0)}}}$$

$$\text{5.}\quad \text{5.1}\quad P(\text{feuille:uc:org}, [\text{à-1}], X)_{\text{global(n+1)}} = \frac{\Sigma_X\, P(\text{feuille:uc:org}, [\text{à-1}], X)_{\text{local(n)}}}{\#\text{dep}_{\text{local(n)}}(\text{feuille})}$$

$$\text{5.2}\quad P(Y, [\text{à-1}], \text{nervure:uc:org})_{\text{global(n+1)}} = \frac{\Sigma_Y\, P(Y, [\text{à-1}], \text{nervure:uc:org})_{\text{local(n)}}}{\#\text{dep}_{\text{local(n)}}(\text{nervure})}$$

$$\text{5.3}\quad P(\text{feuille:uc:org}, [\text{à-1}], \text{nervure:uc:org})_{\text{global(n+1)}} = P(\text{feuille:uc:org}, [\text{à-1}], X)_{\text{global(n+1)}}\; P(Y, [\text{à-1}], \text{nervure:uc:org})_{\text{global(n+1)}}$$

$$\text{6.}\quad P(\text{feuille:uc:org}, [\text{à-1}], \text{nervure:uc:org})_{\text{local(n+1)}} = \frac{P(\text{feuille:uc:org}, [\text{à-1}], \text{nervure:uc:org})_{\text{local(n)}}\; P(\text{feuille:uc:org}, [\text{à-1}], \text{nervure:uc:org})_{\text{global(n+1)}}}{\Sigma_{X,Y}\; P(\text{feuille:uc:X}, [\text{à-1}], \text{nervure:uc:Y})_{\text{local(n)}}\; P(\text{feuille:uc:X}, [\text{à-1}], \text{nervure:uc:Y})_{\text{global(n+1)}}}$$

Table 2: *Extraction of classes for* "feuille à nervure denticulée"

local injection of these preferences in the sentences, re-inforcing the local probabilities. The normalization is given by previous local and global weights for the dependency involving all possible lexical categories associated to each one of the terms considered, and here represented by variables `X` and `Y`.

In dealing with semantic class assignment, the sequence of steps is shown in Table 2, illustrating the computation of the probability that *"feuille"*(`"leaf"`) and *"nervure"*(`"vein"`) are both organs, taking again the dependency labeled [à-1] in Fig. 2:

4. In each sentence, we compute the local probability of this dependency if *"feuille"* (`"leaf"`) and *"nervure"* (`"vein"`) are both organs (`org`). We start from the local weight computed in Table 1, and also the initial preferences the terms involved corresponding to the classes considered[2]. The normalization is given by the probabilities for the possible classes involving each one of the terms considered, without specifying any particular class here represented by variables `X` and `Y`.

5. We calculate this preference at global level, by re-introducing it to the whole corpus locally in the sentences in order to re-compute the weights of all the possible classes in the sentence. We first compute the probability in the whole corpus (5.1 and 5.2) for each term and semantic class, disregarding the right and left context, represented by variables `X` and `Y` respectively. The probability (5.3) is a combination of the two previous ones.

6. After each iteration, we re-inject the previous global weight to obtain a new local one, by re-inforcing the local probabilities. The normalization is done by the addition of the preferences corresponding to the terms and classes involved in the dependency, for all the possible semantic classes considered.

5 Experimental results

We describe some preliminary tests, using the running corpus as guideline. We consider two different quality references. The former, the number of learned elements. Secondly, the computational efficiency on a standard platform. Whatever is the case, these tests are performed in function of the number of iteration learning passes, once we have fixed three thresholds:

- First, the number of the occurrences of a term, that is the number of the governor/governed nodes in the graph of dependencies. This allows us to estimate the validity of the testing frame.
- Second, the percentage for success, showing possible existing relationships between computational loading and efficiency.
- Third, the probability of a dependency being non deterministic, looking to illustrate the impact of ambiguities on the learning task.

that we illustrate in Figs. 4, 5 and 6. As starting point, we take the information compiled for 6 organs, 10 properties and 10 markers.

More in detail, Fig. 4 reflects the execution time for the knowledge acquisition process, considering terms that appear more than 18 times, with a success index of over 90%. We consider here two tests, one related to dependencies whose probability is 1, and the other one focused on dependencies with a probability of over 0'2. The results seem to indicate a linear behavior in the first case and a polynomial complexity in the second one. Intuitively, this conclusion was expected given that knowledge acquisition should be more efficient in dealing with dependencies that are totally guaranteed.

In the same way, the number of learned elements seems to be greater when dealing with high confidence dependencies, as shown in Fig. 5, than when working with the weaker ones included in Fig. 6. Another interesting point is the behavior observed for the different classes learned in Figs. 5 and 6. So, properties, such as

[2] this is fixed by the user, in the case of the term being in a list associated to that class. Otherwise, this probability is obtained as a ratio of the total number of classes considered.

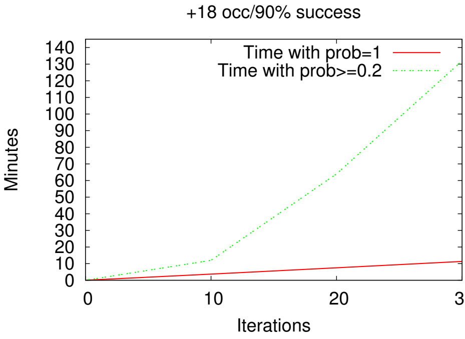

Fig. 4: *Time complexity for the learning process*

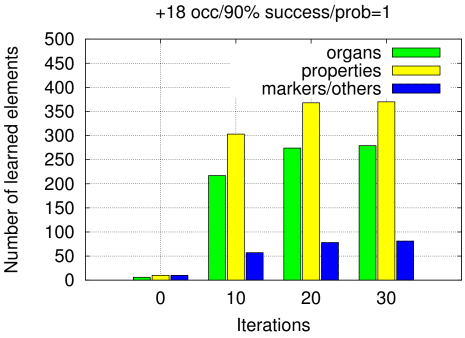

Fig. 5: *Learning dependencies with high probability*

`form` or `color`, are in both cases the classes on which knowledge acquisition runs with greater certainty.

This is also the case with the classes on which term extraction was already defined at lexical level, involving extremely precise linguistic information, as is the case of `organs`. In consequence, knowledge acquisition on these terms is relatively independent of the iterative process and, in particular, of the level of probability considered for dependencies. This is underlined by the asymptotic behavior, when the number of iterations grows and the process converges, showing a similar behavior in both cases.

In the same sense, the asymptotic behavior observed in Figs. 5 and 6 seems to indicate that organs reach a high degree of recognition, depending on the probability of the dependencies considered. As we have seen in our running examples, this is justified by the fact that term extraction on these classes cannot be defined at lexical level, but often relies on the disambiguation of non-deterministic syntactic structures, which concerns the iterative knowledge acquisition process described.

Other marginal categories less involved in term extraction due to the absence of relevant lexical and/or syntactic information, show a closed behavior regardless of the probability considered for dependencies in Figs. 5 and 6. This explains the poor evolution on the number of elements learned in comparison with the results previously obtained on properties and organs.

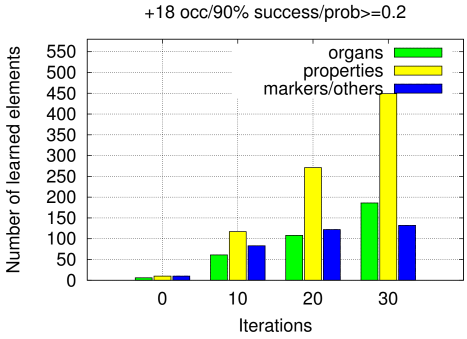

Fig. 6: *Learning dependencies with poor probability*

6 Conclusions

We have introduced knowledge acquisition with a maximum degree of unsupervised tasks. The identification of semantic classes is approached from the detection of similar syntactic contexts around pivot terms. Existing relations between semantic classes are approached from the lexical and/or syntactic patterns connecting them, by using an error-mining technique.

Acknowledgments

This research was partially supported by the Spanish Government under project TIN2004-07246-C03-01, and the Autonomous Government of Galicia under project PGIDIT05PXIC30501PN and the Network for Language Processing and Information Retrieval.

References

[1] M. Alonso, D. Cabrero, E. de la Clergerie, and M. Vilares. Tabular algorithms for TAG parsing. In *Proc. of EACL'99*, pages 150–157, 1999.

[2] J. Bouaud, B. Bachimont, J. Charlet, and P. Zweigenbaum. Methodological principles for structuring an ontology, 1995.

[3] E. de la Clergerie. Dyalog: a tabular logic programming based environment for nlp. In *Proc. of 2nd Int. Workshop on Constraint Solving and Language Processing*, 2005.

[4] E. de la Clergerie. From metagrammars to factorized TAG/TIG parsers. In *Proc. of IWPT'05*, pages 190–191, 2005.

[5] B. Habert, E. Naulleau, and A. Nazarenko. Symbolic word clustering for medium-size corpora. In *COLING*, pages 490–495, 1996.

[6] Z. Harris. *Mathematical Structures of Languages*. John Wiley & Sons, New York, U.S.A., 1968.

[7] C. Jacquemin and D. Bourigault. Term extraction and automatic indexing. *Handbook of Computational Linguistics*, pages 599–615, 1999.

[8] A. Joshi. An introduction to Tree Adjoining Grammar. In A. Manaster-Ramer, editor, *Mathematics of Language*, pages 87–114. John Benjamins Company, 1987.

[9] G. Rousse and E. de la Clergerie. Analyse automatique de documents botaniques: le project biotim. In *Proc. TIA'95*, pages 95–104, 2005.

[10] B. Sagot and E. de la Clergerie. Error mining in parsing results. In *Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 329–336, 2006.