Enhancing the expressiveness of linguistic structures

J. Mora, J. A. Ramos, G. Aguado de Cea

Ontology Engineering Group – Universidad Politécnica de Madrid. Spain
{jmora, jarg, lupe}@fi.upm.es

Abstract. In the information society large amounts of information are being generated and transmitted constantly, especially in the most natural way for humans, i.e., natural language. Social networks, blogs, forums, and Q&A sites are a dynamic Large Knowledge Repository. So, Web 2.0 contains structured data but still the largest amount of information is expressed in natural language. Linguistic structures for text recognition enable the extraction of structured information from texts. However, the expressiveness of the current structures is limited as they have been designed with a strict order in their phrases, limiting their applicability to other languages and making them more sensible to grammatical errors. To overcome these limitations, in this paper we present a linguistic structure named "linguistic schema", with a richer expressiveness that introduces less implicit constraints over annotations.

Keywords: Pattern Matching, Pattern Recognition.

1 Introduction

Text understanding covers a series of tasks such as document classification [13], machine learning [9], information retrieval [3], etc. To perform these tasks, two processes are generally carried out: the recognition of structures and the interpretation of them. In the first one, the aim is to find some specific structures (for example, the pattern *AGENT buys OBJECT* in the text of a web page). Depending on the results found in the search (for example, *AGENT=Pepe* and *OBJECT=flores*, *AGENT=Paco* and *OBJECT=bombones*) the interpretation process triggers the action corresponding to the task performed (learning task, classification task, etc.). In other words, during the interpretation process, the document is classified (for example, G*oods Transactions*), something is learnt (for instance, *Pepe* and *Paco* are instances of *Person*), some information is retrieved (for example, *flores* and *bombones* are goods sold in the Web), etc. Generally speaking, the process of structure recognition is common and independent of the interpretation process, although this process can be instantiated in a battery of structures that might be needed for a later specific interpretation. However, the recognition process itself does not vary. It is in the above mentioned structures on which this work is focused: studying and upgrading their representations and the expressiveness of these representations. This expressiveness will determine the searches: the greater the expressiveness, the more searches can be conducted and the more complex these searches can be. Large scale corpora present greater opportunities in terms of quantity and variety. On a par with these possibilities

they present new challenges with respect to the variety of grammatical constructions used, freedom of language (as opposed to controlled vocabularies), and diversity in topics for the interpretation process. However, these factors increase the ambiguity in the recognition of structures in the text. Therefore, the language of representation for these structures and its components, operators and hypotheses is of paramount importance.

Although recognition structures are widely used, and many examples with different interpretations can be found, it is not so easy to find a specification of the language in which these linguistic structures are expressed, nor the formalization used to express the restrictions involved. Furthermore, these representations of structures have been focused more on human legibility than on machine interpretation, although computational systems need a formal form of representations to work. In fact, these systems use a formal representation, but this is implicit and has not been fully explained. For that reason, sharing the structures, defined following a specific representation, is not a trivial issue.

In this paper we present a well defined proposal of formal representation to express linguistic structures of recognition. For the purpose of this work, we have named them "linguistic schemas", in which the meaning of all the elements appearing in the structures is made explicit. Moreover, a formal representation of these linguistic schemas, which is also interpretable by a computational application, is specified. The main aim is to provide these recognition structures with the capability of being reused and shared by different tools and systems, and to allow this formal representation to be explicit, well defined and computationally interpretable. This proposal aims at solving the complex problem of expressiveness in linguistic structures for NLP.

Thus, section 2 presents the representation specifications of linguistic structures. Section 3 offers a view of the linguistic scenario in which we can find the need for these new linguistic structures. The representation of the linguistic schemas is presented in section 4 and they are exemplified. Section 5 analyzes the expressiveness of the existing recognition structures comparing them with the new one developed and presents the results and future work. Finally references are also included.

2 Linguistic structures in use

Linguistic patterns are used in Computational Linguistics to understand natural language texts. Among the most outstanding projects it is worth noting the program PHRAN (PHRasal Analysis) [2, 16], which tackles the implementation of an approach based on knowledge. PHRAN deals with pattern-concept pairs (PCPs), whose linguistic components are phrasal patterns that may present different abstraction levels. This means that the pattern may be composed by a word, a literal string, as "Digital Equipment Corporation" or a general phrase as "<component> <send> <data> to <component>", enabling any object with the semantic category "component" to appear in the first and last position, any verbal form of "send" to appear in the second position, the word "to", in the fourth position, etc. There is also a conceptual template associated to each phrasal pattern, in which the meaning of the phrasal pattern is described.

In the field of information acquisition from machine readable dictionaries (MRDs), Hearst [5] developed a set of lexical-syntactic patterns restricted to identifying hyponymy relations in texts. Kim and Moldovan [7] created the *FP-structures* (*Frame-Phrasal pattern structure*), which are pairs composed by a frame of meaning and a phrasal pattern, as the one used in PALKA (*Parallel Automatic Linguistic Knowledge Acquisition System*).

More recently, the development of systems for automatic knowledge extraction has generated a substantial amount of works focused both on representations and systems. A detailed analysis can be found in the compilatory study by [17].

All in all, the lexical-syntactic patterns are generally expressed by means of operators in the *Backus-Naus Form* (BNF) in order to compose regular expressions in context-free grammars. Jacobs *et al*. [6] make this explicit when they take the following operators to express lexical-semantic patterns:

Lexical features that can be tested in a pattern: token "name" (ej. "*AK-4T*"), root (ej. "*shoot*"), lexical category (ej. "*adj.*")
Variable assignment from pattern components: ?X =
Logical combination of lexical feature tests: OR, AND, NOT
Wild cards: $ **-** 0 or 1 token, *** -** 0 or more tokens, **+ -** 1 or more tokens
Grouping operators: <> for grouping, [] for disjunctive grouping
Repetition: * - 0 or more, + - 1 or more
Range: *N - 0 to N, +N - 1 to N
Optional constituents: `{}` - optional

Linguistic patterns, be they lexical-syntactic, semantic or, as in the case of PALKA, structures of phrase frames, are always ordered sets of components that express characteristics or constraints on the phrase elements. In every case, the phrase element order and the pattern component order will be the same, even if not explicitly indicated, as all of them are patterns for English, a language with a strict phrase order [12] compared to other Romance languages, for instance. However, when the texts processed by the system are written in a natural language without these constraints, these patterns, which are equivalent to regular expressions, do not fulfill the objectives; then, a wider representation enabling not to specify the order in which the phrase elements should appear, is required. Therefore one of these wider patterns will match the same phrases as a set of ordered patterns, which correspond to different permutations of the same pattern components.

This problem is partially solved by Hazez [4], as he takes morphemes, words, grammatical categories or a syntactic pattern as linguistic patterns. These linguistic patterns are managed as segments to which certain set operators, such as union and intersection, and other operators that express position and content are applied.

Linguistic patterns based on annotations can be found in other cases, as in Specia and Motta's work [14], but the annotations used are always simplified. Thus, in the following example taken from Specia and Motta, based on the relation extraction between phrasal components, and performed by the system Minipar [8], everything is simplified to a triplet over which the patterns are established: <noun_phrase, verbal_exp., noun_phrase>. In this same line, syntactic patterns are applied to disambiguate [11]. Table 1 contains a comparison of the pattern features in these approaches.

Table 1. Comparison of pattern features

Phrasal Pattern	**Lexical-syntactic pattern**	**(Hazez)**	**(Specia and Motta)**
Elements			
literal string, general phrase, semantic category identifier	token "name", root, lexical category, conceptual category, variable	variables, morphemes, words, grammatical categories, linguistic pattern	triplet of token annotations
Operators			
order (in general phrase)	OR, AND, NOT, $, *, +, <>, [], *N, +N, { }, order (in sintaxis)	set operators (∪,∩, etc.), position, content (⊃,⊂, etc.)	
Hypothesis			
	closed world		search triplets

3 Linguistic scenario

The purpose of this work is the understanding of Spanish texts annotated electronically by software tools. In order to enable the automatic application of these patterns to large scale corpora, we have established some constraints over the phrases in several levels, specifically in orthographical, morpho-syntactic, and syntactic levels. However, the possibility of including annotations of any other level (such as semantic, pragmatic or discursive) remains open.

As for the works about annotation and creation of linguistic patterns to extract information from texts in Spanish, the initiatives grow in number and importance as the multilinguality significance increases in the Internet. In the framework of the European project SEKT[1], one of the use cases was focused on the Spanish legal terminology for the creation of ontologies in the legal domain. For this task Hearst's taxonomic relation patterns [5] were translated into Spanish and new patterns were created with the purpose of using the knowledge obtained to enrich ontologies [15].

Related with knowledge extraction for ontology enrichment and population in Spanish we can find another classification attempt in Álvarez de Mon y Rego y Aguado de Cea [1]. These authors extended Hearst's patterns by focusing on certain patterns with classification verbs such as *clasificar*, *figurar*, *distinguir* or *dividir*, that allow a more complete extraction of concepts hierarchically related.

Nica's *et al.* [11] work about desambiguation has been also applied to Spanish for extracting syntactical-semantic patterns (formalizations of the argument-predicate structure related with a verb) from an annotated corpus [10].

We decided to represent linguistic structures in XML format to work computationally with these structures in an easier way as XML is the language most widely used for knowledge representation and many tools can process it. However, files in XML cannot be easily read by humans because of the verbosity of its syntax.

[1] http://www.sekt-project.com/

4 Linguistic schemas

A linguistic schema is a set of constraints over the tokens of a phrase (token contraints) and over the relations between these tokens (phrase constraints). Token constraints are expressed as a set of values of characteristics of annotations of a token. Phrase constraints are expressed using operators (optimality, grouping, etc.) over token constraints or other phrase constraints.

As previously stated, the complete representation of the schemas is stored in XML files for an easier computational processing. Although these files can be read by a person, this task is rather tedious and can be untractable if the number or size of the schemas grows significantly.

For this reason a shortened and user-friendly annotation is defined. This annotation may serve as a mnemonic of the schemas that appear in a file. It does not comply with the XML conventions and may not contain all the information available in the schema. However, the annotation is much easier to read, and, if used correctly, it may identify the schema that is referred to without any short of ambiguity.

Furthermore, this notation has been extended with additional operators, which are not present in the XML notation, to increase the expressiveness and improve the shortness. These operators are replaced by combinations of the operators available in the XML notation. As an example, the optionality operator (see section 4.2) applied to a token would be replaced with a disjunction between this token and the negation of the same token.

A brief summary of the notation proposed (for a friendly representation) is:

Token constraints (Elements): constant (ej. "*shirt*"), identifier (ej. "*ANIMAL*")

Phrase constraints (Operators):

Order operators: A ⊕ B – A appears before B, A + B – A appears immediately before B

Disjunction operators: A | B – A and B can appears, A / B – A or B can appear

Grouping operator: ( ) – group

Repetition operator: * - 1 or more times

Negation operator: ¬ A – A doesn't appear

Optionality operator: [ ] – optional

General hypothesis: Open world

4.1 Terms

For the purposes of this work, a term in a linguistic schema is the set of constraints, in other words, the set of elements applied to one single token. In the user-friendly syntax, these terms may be displayed with two different types of symbols: constants and identifiers.

- Constants are words written as they appear in the text, for example *clasifica*.
- Identifiers are used to retrieve values instead of restricting them, and they appear as strings in uppercase, for example "ACTOR".

Terms with identifier and/or lemma

In those cases in which an identifier ("ACTOR") or a lemma (*clasificar*) is specified this will be shown in the set of constraints of a token.. For example, when a token has as a constraint the lemma *clasificar*, and its morpho-syntactic value is "main verb", only the lemma will be shown. If both data about the same term are used in the information, then the identifier will be shown. For example, when a token has as a constraint the lemma "*clasificar*" and as text the identifier "CONJUGATED_FORM", then only the identifier "CONJUGATED_FORM" will be shown. If a set of identifiers is specified for a token, then the identifier whose value has previously appeared will be used, according to the annotation standard used. For instance, if two identifiers are assigned to a token, such as the values of gender ("GENDER") and syntactic function ("FUNCTION"), only the former will be shown, i.e., "GENDER".

It is possible to use the identifier to refer to any of the non constant terms. For instance, the next schema can be written using the identifiers A, B, C and Z:

```
A + come + B + y + C + en + Z
```

This schema would match a phrase such as "*Pepe come pan y chocolate en el patio de la escuela*", and in this matching the identifiers will take the values corresponding to this specific phrase: *A=Pepe*, *B=pan*, *C=chocolate*, *Z=patio*.

Terms with the category specified in any annotation level

For those terms for which no identifier or lemma are specified, but the value of, at least, a category in an annotation level is defined, the name of that category will be shown. Taking as reference the previous example, the values "verb" or "direct object" will appear instead of "come" and "B".

If an abbreviated form is specified for any category in the standard used[2] and possibly with information about additional attributes (for example "Fused_Prep-At" for "Fused Preposition-Article"), then the most specific abbreviated form will be shown for each annotation level, being the most specific form the one that includes more information about the additional attributes. Thus, when we want to identify a token that is an ordinal pronoun, but of which we do not want to obtain any other information, its lemma or value for any other category (as in the phrase "el primero es el grande") is described as "Ordinal_pronoun" as we only want to restrict this word to this type of pronoun.

4.2 Operators

As previously mentioned, operators define the relations among the different parts of a schema. It is necessary to point out that the order in which the parts of the phrase must appear is not specified by the element appearance order in the schema; therefore, if it is necessary to set this order, then it must be specified explicitly. This can be done with two symbols:

- With the symbol '+': the expression "*symbol1* + *symbol2*" means that "*symbol2*" must appear immediately after "*symbol1*".

[2] http://pln.oeg-upm.net/annotation/ontotag

- With the symbol '⊕': the expression "*symbol1* ⊕ *symbol2*" means that "*symbol2*" must appear after "*symbol1*", immediately or not.

There are other symbols besides the previous ones which express different relations. These symbols are the following:

- **'*'**: expresses repetition.
  For example, "*symbol**" means that "*symbol*" may appear more than once.
- **'('** and **')'**: groups several symbols.
  For example, "(*symbol1* + *symbol2*)*" means that "*symbol1*" may appear several times, all of them followed by "*symbol2*".
- **'['** and **']'**: means that whatever is between both square brackets is optional.
  For example, "[*symbol*]" means that "*symbol*" may appear or not in the phrase.
- **'|'**: means that either what is in the left side or what is in the right side must appear.
  For example "*a*|*an*" means that "*a*" or "*an*" must appear.
- **'/'**: means that either what is on the left side or what is on the right side must appear, but not both of them.
  For example "*a*/*an*" means that "*a*" or "*an*" must appear, but not "*a*" and "*an*" at the same time.
- **'¬'**: means that the next element must not appear in the phrase. When combined with the symbols + and ⊕, it may indicate that the said symbol must not appear in some specific positions of the phrase.
  For example, "¬*symbol*" means that "*symbol*" may not appear in the phrase.

Examples of linguistic schemas

To show the versatility and possibilities of the linguistic schemas we include some examples, expressed in the user-friendly notation.

We want to identify who buys things to María, and which those things are. Hence, we express these constraints in a linguistic schema setting the main verb ("*compra*") and the indirect object ("*a María*"). The rest of the phrase and the order of appearance are not restricted. These constraints may be expressed with the next linguistic schema:

```
X compra Y a + María
```

This schema would match phrases like "*Pepe compra a María flores*", "*Pepe a María flores compra*", "*Pepe compra flores a María en domingo*", "*A María Pepe le compra flores*" and "*Juan a María compra bombones de licor en Santander*".

It is worth mentioning that the previous schema would be equivalent to the next one, since linguistic schemas have no implicit order, as it happens in the case of lexical-syntactic patterns. Also, the name assigned to the identifier does not change the recognition capabilities of a linguistic schema:

```
SOMEONE a + María SOMETHING compra
```

This schema would match with exactly the same phrases as the previous one. However, it would take 24 lexical-syntactic patterns (P(4,4) = 4! = 24) to match the same phrases using patterns, as there are four pattern components in the previous example, (1) SOMEONE, (2) a+María, (3) SOMETHING and (4) compra. Moreover,

these patterns could also have additional elements in the phrase, resulting in a larger list of lexical-syntactic patterns.

Because of this combinatorial explosion and the open world assumption, processing a schema requires more computational power than a pattern. However, our proposal for a schema represents a set of patterns in a more compact way, enabling a further optimization and more efficient algorithms.

The application of these linguistic schemas to Spanish does not mean that they cannot be used for other languages. For the lexical-syntactic pattern

```
X buys Y for María
```

the equivalent linguistic schema would be:

```
X + buys + Y + for + María
```

An example can be seen in pln.oeg-upm.net/process/linguisticschemas.

5 Comparison and discussion

Once we have described and exemplified the notation proposed, we will compare the expressiveness of our notation with the lexical-syntactic pattern notation, accepted by Jacobs *et al*.[6].

The first point is that the notation we propose assumes the open world assumption. This assumption means that everything that is not described in the schema is not restricted, thus, it can appear or not.

Table 2. Comparison of Lexical-syntactic patterns and Linguistic schemas

Lexical-syntactic patterns	**Linguistic schemas**
Lexical features:	
token "name"	text value
Root	lemma value
lexical category	values of these categories
conceptual category	
Combination of lexical features:	
OR	operator **'|'**
AND	implicit
NOT	operator **'¬'**
Wild cards:	
$, *, +	These operators are unnecessary taking into account the open world hypothesis
Variable assignment from pattern components:	
?X =	identifiers
Grouping operators:	
<>	**'( )'**
[]	combination of **'( )'** and **'/'**
Repetition:	
*	combination of **'[ ]'** and **'*'** inside

+	**‘*’**
Range:	
*N, +N	Extensional representation
Optional constituents:	
{}	**‘[ ]’**

In the Table 2, we can see the notation with the regular expressions used by Jacobs *et al*. (left column) and the correspondences to our notation (right column).

In the case of range, the term “Extensional representation” involves iterating n times the term optionally. That is, it can be represented, but it does not have an operator or a sign that compresses this expression.

Besides covering completely the expressiveness of the previous notation, the new notation contributes the following functionalities:

- It takes into account the values of all the characteristics of the annotations (not only lexical and conceptual categories).
- It includes identifiers to be used in any value of the annotation (not only the four lexical characters provided by Jacobs).
- It includes operators of exclusive disjointness ‘/’ in and out of the group.
- It includes operators of order ‘+’ and ‘⊕’.
- It allows applying these operators to sub-schemas (not only to the four lexical characters dealt in Jacobs’).
- The open world assumption allows ignoring which tokens can appear or not in phrases. For this reason wild cards are not necessary.

We consider that this comparison shows that any lexical-syntactic pattern expressed in traditional notation can be expressed in terms of the notation proposed as a linguistic schema.

This representation permits, first, describing constraints about text annotations and, second, dealing with other previously created linguistic structures.

With these characteristics we have designed linguistic structures which describe phrases in languages that do not have a rigid morpho-syntactic order, such as Spanish.

As future work, the implementation of an assistant (already designed) for editing schemas will make linguists work easier and will contribute to a greater automatization.

The assistant should allow the definition of many schemas comfortably. Presumably, this combination of quality and quantity should allow a greater automation of NLP tasks, improving the results when processing large scale corpora.

7 Acknowledgments

This work has been supported by the project *BabeLData* (TIN-2010-17550), funded by *INIA*, under *Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica* (I+D+i) of *Ministerio de Ciencia e Innovación*.

8 References

[1] Álvarez de Mon y Rego I, Aguado de Cea G (2006) The phraseology of classification in Spanish: integrating corpus linguistics and ontological approaches for knowledge extraction. BAAL/IRAAL Joint Int. Conf., Ireland.

[2] Arens Y (1986) *CLUSTER: An approach to Contextual Language Understanding*. Ph.D thesis, Univ. of California at Berkley, 1986.

[3] Baeza-Yates R, Ribiero-Neto B (1999) *Modern information retrieval*. Addison Wesley Longman, Essex, England.

[4] Hazez SB (2001) Linguistic pattern-matching with contextual constraint rules. *IEEE International Conference on Systems, Man, and Cybernetics*. Vol. 2. Pages: 971-976. Tucson, AZ, USA, October 7th-10th, 2001.

[5] Hearst MA (1992) Automatic Acquisition of Hyponyms from Large Text Corpora. *COLING-92*. Nantes, 23-28 August, 1992.

[6] Jacobs PS, Krupka GR, Rau LF (1991) Lexico-semantic pattern matching as a companion to parsing in text understanding. *In Fourth DARPA Speech and Natural Language Workshop*, pp. 337-342, 1991.

[7] Kim JT, Moldovan DI (1993) Acquisition os Semantic Patterns for Information Extraction from Corpora. *Ninth Conf. AI applications*, 1993.

[8] Lin D (1993) Principle based parsing without overgeneration. *31st ACL*, Columbus, pp. 112-120. 1993.

[9] Mann T (1993) *Library research models*. Oxford University Press, NY.

[10] Navarro B, Moreno-Monteagudo L, Martínez-barco P (2006) Extracción de relaciones sintagmáticas de corpus anotados. *Procesamiento de Lenguaje Natural*, ed. SEPLN, nº 37, septiembre 2006, pp: 59-66

[11] Nica I, Martí NA, Montoyo A, Vázquez S (2004) Intensive Use of Lexicon and Corpus for WSD. *Procesamiento de Lenguaje Natural*, ed. SEPLN, nº 33, septiembre 2004, pp: 147-154

[12] Quirk R, Greenbaum S (1977) *A University Grammar of English,* London, Longman.

[13] Sebastiani F (2002) Machine learning in automated text categorization. ACM Computing Surveys, Vol. 34, No. 1, pp. 1–47. 2002.

[14] Specia L, Motta E (2006) A hybrid approach for extracting semantic relations from texts. *2nd Workshop on Ontology Learning and Population en COLING/AC*L 2006. Sydney, Australia. July 22nd, 2006.

[15] Völker J, Vrandečići D, Sure Y (2006) SEKT Project *D3.3.3 Data-driven Change Discovery.* SEKT Project.

[16] Wilensky R, Arens Y (1980) PHRAN: A Knowledge-Based Natural Language Understander. *18th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA. June 1980.

[17] Zhou N, Zhou X (2004) Automatic Acquisition of Linguistic Patterns for Conceptual Modeling. *Course "INFO629: Concepts in Artificial Intelligence*". Drexel University, Fall 2004.