Natural Language Engineering **23** (1): 131–153. © Cambridge University Press 2015
doi:10.1017/S1351324915000297

A method based on rules and machine learning for logic form identification in Spanish†

F. MARTÍNEZ-SANTIAGO, M. C. DÍAZ-GALIANO, M. Á. GARCÍA-CUMBRERAS and A. MONTEJO-RÁEZ

Department of Computer Science, Universidad de Jaén, Paraje Las Lagunillas, s/n, 23071, Jaén, Spain
e-mail: dofer@ujaen.es, mcdiaz@ujaen.es, magc@ujaen.es, amontejo@ujaen.es

(*Received 7 March 2014; revised 14 July 2015; accepted 15 July 2015; first published online 24 August 2015*)

Abstract

Logic Forms (LF) are simple, first-order logic knowledge representations of natural language sentences. Each noun, verb, adjective, adverb, pronoun, preposition and conjunction generates a predicate. LF systems usually identify the syntactic function by means of syntactic rules but this approach is difficult to apply to languages with a high syntax flexibility and ambiguity, for example, Spanish. In this study, we present a mixed method for the derivation of the LF of sentences in Spanish that allows the combination of hard-coded rules and a classifier inspired on semantic role labeling. Thus, the main novelty of our proposal is the way the classifier is applied to generate the predicates of the verbs, while rules are used to translate the rest of the predicates, which are more straightforward and unambiguous than the verbal ones. The proposed mixed system uses a supervised classifier to integrate syntactic and semantic information in order to help overcome the inherent ambiguity of Spanish syntax. This task is accomplished in a similar way to the semantic role labeling task. We use properties extracted from the AnCora-ES corpus in order to train a classifier. A rule-based system is used in order to obtain the LF from the rest of the phrase. The rules are obtained by exploring the syntactic tree of the phrase and encoding the syntactic production rules. The LF algorithm has been evaluated by using shallow parsing with some straightforward Spanish phrases. The verb argument labeling task achieves 84% precision and the proposed mixed LFi method surpasses 11% a system based only on rules.

1 Introduction

Logic Forms (LF) is a flat, scope-free, first-order logic (FOL) representation that embeds lexical and syntactic information (Rus 2002, 2004b). A more philosophical approach is focused on how grammatical structure is related to logical form, how meaning is related to truth, and how human concepts are related to linguistic understanding (Russell 1914; Lepore and Ludwig 2001; Pietroski 2009). Our interest

†This work has been partially funded by the ATTOS project (TIN2012-38536-C03-01) from the Spanish Government and the AORESCU project (TIC 07684) from the Andalucía Government.

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

is mainly related to the first point only, how we could obtain automatically the logical form from the grammatical structure, which is called Logic Form identification (LFi).

In recent years, LFi has been used to express texts in natural language in a formal way, with a good balance between the complexity of the model and its expressiveness (Delmonte and Rotondi 2012). LFi systems are usually based on rules extracted from the syntactic analysis of the text. In these cases, the system is limited to the knowledge contained in the syntactic rules and to syntactic ambiguity. Furthermore, the codification of the rules is manual.

Since, LF is a syntactic representation and there are other syntactic representations such as dependency grammars, why should we emphasize LF? Because LF is FOL and this eases reasoning with natural language:

- It is straightforward to add new knowledge by means of new rules. For example, it is possible to add rules in order to manage ambiguous constructions or semantic information.
- If reasoning about the language is required, then it is possible to apply an automated theorem prover for FOL.

A good example to illustrate both advantages is COGEX (Moldovan and Rus 2001). COGEX is a modified version of the OTTER (McCune 1994) logic prover that has been adapted for natural language processing (NLP). Thus, COGEX integrates homogenously syntactic knowledge and knowledge of the world by means of LF.

The automatic derivation of LF is far from being a trivial problem. Resources such as eXtended WordNet (Rus 2002) contain several errors such as free variables and/or predicates without any relation with any other predicates in the definition, fictitious arguments, compound words which are identified differently in different contexts and so on (Delmonte and Rotondi 2012). It is important to note that, in our paper, the term 'predicate' always refers to a FOL predicate, not to a grammatical component. Delmonte and Rotondi evaluated eXtended WordNet and they obtained an average error rate of 30%. These problems increase in the case of languages with multiple valid syntactic parsing trees due to syntax flexibility, as with Spanish.

This paper proposes a new method that allows the inclusion of well-known semantic role detection and classification techniques to improve the LFi of verbs. In addition, machine learning and the integration of different information sources (tagged corpora) make possible to decrease the number of manually codified rules. In short, we propose a mixed method for Spanish, and the application of semantic role labeling (SRL) in order to obtain the LF of verbs.

The similarity between the classifier of the mixed method and the SRL task is significant, but identifying logical roles is a relatively simpler task compared to the shallow semantic parsing which includes two subtasks: argument classification and argument identification. SRL is similar to LFi in the sense that both of them consist of the detection of the semantic arguments associated with the predicate or verb of a sentence and their classification into their specific roles. This is the reason why we

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

think that SRL is a more suitable approach to obtaining the LF identification for phrases written in Spanish and the other Romance languages.

The rest of the paper is organized as follows. First, the LFi task is presented: its definition and main approaches, as well as some problems of its application to a language with flexible grammar, like Spanish. Sections 2 and 3 describe some applications of LFi and we make a brief introduction to the SRL task and its application to LFi. Sections 4 and 5 describe the proposed method and the experiments. Finally, some conclusions and future work are reported.

2 Logic form identification

LF are simple, FOL knowledge representations of natural language sentences formed by the conjunction of concept predicates related through shared arguments. Each noun, verb, adjective, adverb, pronoun, preposition and conjunction generates a predicate. LF can be decorated with word senses to disambiguate the semantics of the word. There are two types of variables: events are marked with '*e*', and entities are marked with '*x*'. The shared arguments connect the subjects and objects of verbs and prepositions together. For example, the sentence '*Earth provides the food we eat every day*.' could be represented as

$$Earth_N(x_1)\ provide_V(e_1,x_1,x_2)\ food_N(x_2)\ we(x_3)\ eat_V(e_2,x_3,x_2;x_4)\ every_J(x_4)\ day_N(x_4)$$

where, _N indicates noun, _V indicates verb, _P indicates pronouns or determiners.

The identification of LF, based on FOL predicates, aims to obtain a formal representation of natural language between syntactic and semantic levels. Thus, the LF that we use is a flat, scope-free FOL representation that embeds lexical and syntactic information. The LF of a sentence is the conjunction of individual predicates, where the relationships among them are expressed via shared arguments. Predicates are generated for all content words such as nouns, verbs, adjectives and adverbs. Furthermore, in order to maintain a simple model, some simplifications are performed. Inflections of gender, number, time, determiners, negations and auxiliary verbs are ignored. For instance, the Spanish phrase '*Juan cuenta cuentos a su hijo cada noche de verano*' ('*Juan tells stories to his son every summer night*') would be represented as

$$Juan_N(x_1)\ contar_V(e_1,x_1,x_2,x_3;x_4)\ cuento_N(x_2)\ a_P(e_1,\ x_3)\ su_P(x_3)\ hijo_N(x_3)\ cada_J(x_4)\ noche_N(x_4)\ de_P(x_4,\ x_5)\ verano_N(x_5)$$

A relevant property of this model is that the arguments always appear in the same order, as in the phrase, so the application of inference systems over the knowledge base is easier. For instance, working with verbs, the first argument represents the proper verb, the second argument is the subject of the action, the next one is the direct object, then the indirect object. Arguments after semicolon ';' are adjuncts (arguments related to time and space which are not mandatory to convey the meaning of an utterance).

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Table 1. *Some logic form transformations*

Rule	Transformation	Applied over	Result
PP→P NP	prep/P name/NP → prep(-,x) name(x)	en verano (*in summer*)	en(_,x_1) verano(x_1)
NP→NP PP	name/NP prep/PP(_,x_1) → name(x_2) prep($x2$,$x1$)	noche en verano (*night in summer*)	noche(x_2) de(x_2, x_1) verano(x_1)
NP→JJ NN	name/NP adjective/JJ → name(x) adjective(x)	blanca nieve (*white snow*)	blanca(x_1) nieve(x_1)
NP→NP CC NP	name/NP conjunction/CC name/NP → conjunction(e, x, y) name(x) name(y)	tierra y mar (*land and sea*)	y(e, x_1, x_2) tierra(x_1) mar(x_2)

2.1 Logic Form identification approaches

The usual way of obtaining the LF of the text is by applying rules derived from the syntactical structure of the phrase. From this point of view, such rules are based on syntax. These rules have two parts, the head of the rule and the tail:

- The head of the rule is a syntactical pattern. For example, the head [*adjective name*] matches with a specific rule in order to translate such syntactical construction to an equivalent LF.
- The tail of the rule is the transformation from the recognized pattern to a (maybe incomplete) equivalent logical form. For example, if [*white_N snow_J*] matches with [*adjective name*] rule then the system recognizes such a matching and it generates the LF [*white_J*(x_1) *snow_N*(x_1)].

Some rules are shown in Table 1. Other rules and their descriptions will be shown later in Section 4.

There are two common approaches to obtaining such rules: those based on constituency trees and those based on dependency trees. The two approaches are based on rules, but systems using dependency trees obtain rules more easily (Wenner 2007; Agerri and Peñas 2010) than those based on constituency trees (Rus 2002). Both approaches require that each syntactic form has a rule developed by a human expert. The number of necessary rules does not only depend on the desired recall but on the complexity of the language. For instance, a syntactically relatively simple language like English needs only ten rules to cover 90% of the WordNet glosses (Moldovan and Rus 2001).

In addition, a system based exclusively on rules requires a new rule for every exception. For example, the elliptical subject is usually labeled by the parser, but if the syntactic analyzer does not tag it, it must be considered as an exception. In our method, the elliptical subject is detected by applying some simple heuristics, like the absence of a noun phrase to the left of the verb (in active phrases). Note that this heuristic is not infallible, but it works because it requires, first, to mark a false

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

positive that, first, the syntactic parser should wrongly label a subject and, second, that the subject should be elliptic. This is possible, but rare.

Note that these systems require precise syntactic analyzers, and they are not always available for every language. For such cases, shallow parsing is applied instead of syntactic parsing. If grammatical roles are not available then it increases the ambiguity of the rules, especially for languages with a more complex syntax than English. Whereas, in English the different parts of a phrase always have a fixed position that depends on its type, in Spanish this is not true. Identification in Spanish of the arguments of the verb is not a suitable task to be resolved by using rules, because word order is not fixed and it can be influenced by syntactic and pragmatic factors, including verb type, animacy, weight, and information structure (Alfaraz 2012). Basic word order in Spanish is Subject-Verb-Object SVO (Suñer 1982) but other word orders are possible. For example, the phrase 'Mary wants a dog' can be translated to Spanish as any of the following active sentences:

- *Mary quiere un perro* (SVO order).
- *Un perro Mary quiere* (OSV order).
- *Un perro quiere Mary* (OVS order).
- *Quiere Mary un perro* (VSO order).
- *Quiere un perro Mary* (VOS order[1]).
- *Quiere un perro* (VO order, the subject is omitted if the context of the phrase gives information about who wants the dog).
- *Un perro quiere* (OV order, the subject is omitted).

In order to detect the subject constituent Rus (2002) proposes the rule 'S→ NP VP NP' but this rule only works correctly in the first translation. Thus, LFi for the classification of Spanish phrase constituents requires not only syntactic information encoded into rules but also morphological and semantic information. Note that state-of-the-art syntactic parsers give grammatical functions of the constituents, which eases the derivation of the LF of the sentence.

The most usual way to obtain the LF of a text is by using a set of manuscript rules. The alternative are systems based on dependency trees. Ahn *et al.* (2004) traverses the dependency structure of the phrase, turning part-of-speech tagged lexical items into LF predicates, creating referential variables for nouns and verbs, and using dependency labels to order the arguments at each node. A similar approach is proposed in Anthony and Patrick (2004).

Systems based on dependency trees (Figure 1) use a representation more similar to the LFi structure than those using constituency trees. It is relatively straightforward to turn dependency structures into LFs (Agerri and Peñas 2010) but even in such cases it is required to encode additional knowledge by using rules.

[1] VSO and VOS orders are not usual but may occur.

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

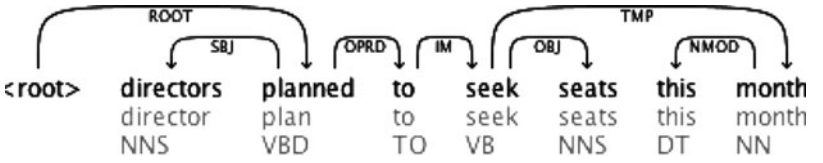


Fig. 1. An example of a dependency tree (source: mate-tools using the approach described in Johansson and Nugues (2008)).

3 Related work

3.1 Logic forms identification applications

The trace of the FOL proofs provides a basis for generating humanly comprehensible interpretation justifications. Thus, LFi is applied in several NLP tasks where the understanding of the text is relevant. A few examples are listed below.

Moldovan *et al.* (2007) implements a question answering system which integrates several sources of knowledge such as WordNet, Suggested Upper Merged Ontology (SUMO) and handmade rules about world knowledge. The whole of the knowledge is managed by using COGEX, a logic prover based on LF. Todorova (2009) proposes a QA system based on LF and the codification of common sense knowledge applied to simple domains.

COGEX has been used for other NLP tasks such as textual entailment (Fowler *et al.* 2005) and automatic answer validation (Tatu *et al.* 2007). In a similar way, Ferrández *et al.* (2007) applies LF to both tasks by inferring the associated LF of a sentence through applying NLP rules to the dependency tree. However, the LF used for COGEX are obtained mainly by using a rule-based approach.

Muñoz-Terol, Martínez-Barco and Palomar (2007) uses LFi in order to modify the weight of the topics terms in a cross-lingual speech retrieval system. The document collections are transcriptions of spoken news and the system aims to find news about several topics. Each topic term in the LF can modify its term weight according to its predicate type and the relationships between all the predicates in the LF of the topic. For example, LFi helps to identify grammatical utterances that have a circumstantial behavior in the sentence, so the nouns in the utterance have a greater weight.

All these previous systems use logical forms of the same type as our proposal.

LG-Soar (Tustison 2004) is a system capable of parsing eligibility criteria in clinical trials and outputting a semantic representation using LFi. This software aims to help information extraction tasks.

Watson (Ferrucci *et al.* 2010; McCord, Murdock and Boguraev 2012) is a computer system that competed at the human champion level in real time on the American TV Quiz show, Jeopardy! The extent of the challenge included fielding a real-time automatic contestant on the show, not merely a laboratory exercise. Watson uses logical form generation for knowledge representation and reasoning related tasks.

Nguyen and Shimazu (2014) and Nakamura *et al.* (2010) have developed a system for translating legal sentences written in Japanese, English and Vietnamese

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Table 2. *An example. SRL for the verb* 'to provide' *in the sentence* 'the Earth provides the food we eat every day'

Noun phrase	Noun phrase type	Role
the Earth	subject	Supplier
the food we eat every day	object	Theme

into logical formulae. The system basically converts words and phrases in a target sentence into predicates in a logical formula, which is used as a kind of interlingua.

Finally, there are several initiatives used in order to codify ontologies such as WordNet to LF (Harabagiu, Miller and Moldovan 1999; Agerri and Peñas 2010). Thus, these resources are available for reasoners based on LF such as COGEX, whenever the LF formalism is the same for the ontology and the reasoner.

3.2 Semantic role labeling

The aim of the SRL task is the detection and classification of the semantic relation between the verb and the other constituents of the phrase, with relations such as agent, patient, instrument, . . . [2](see Table 2).

Gildea and Jurafsky (2002) distinguish between role detection and classification, since a real system that tries to identify the semantic role of a verb must disambiguate it, because the sense of the verb determines these roles. In a simple model, it is given the role detection, so the system has only to define the role type. They propose a model based on learning with six properties per role, and these properties are used to relate the syntax and the semantic of each role.

SRL has received much attention (Màrquez *et al.* 2008; Palmer *et al.* 2010), partly because of its potential to improve applications that require broad coverage semantic processing. Thus, the interest in SRL has been invigorated for several years by the Conference on Natural Language Learning (CoNLL) and the SemEval competitions.

4 The mixed method

The aim of this study is to develop a mixed LFi system based on rules and a new module based on techniques drawn from SRL. This new module is intended to decide the way that the arguments of the LF of a verb are filled. Such arguments will be every variable assigned to every constituent of the sentence. Thus, given the example of Section 2:

$Juan_N(x_1)\ contar_V(e_1,x_1,x_2,x_3;x_4)\ cuento_N(x_2)\ a_P(e1,\ x_3)\ su_P(x_3)\ hijo_N(x_3)$
$cada_J(X_4)\ noche_N(x_4)\ de_P(x_4,\ x_5)\ verano_N(x_5)$

[2] Role names of the example are taken from FrameNet, available at https://framenet.icsi.berkeley.edu [27/05/2014].

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

The module based on SRL will decide that x_1 is the second argument of *contar_V*, that x_2 is the third argument, and so on. The arguments of the logical form of the verb are always subject, direct object, indirect object and other syntactic functions. Therefore, it is necessary to group different roles with the same argument of the logical form of the verb, such as is depicted in Table 6. The reason is because a given argument of a predicate could be relative to several roles. LFi does not require to identify such semantic distinctions. For example, given the sentence the LF of a verb, the subject is the second argument of such a LF whereas from the point of view of SRL, the same subject could be labeled as 'Reader' ('I read a book'), 'Player' ('I play football'), or anything else.

In short, syntactic rules are applied in order to obtain the LF of non-verbal constituents: nouns, adjectives, adverbs, pronouns, prepositions and phrases such as noun phrases, adjective phrases, adverbial phrases, prepositional phrases, coordinated phrases. . . . On the other hand, a classifier inspired by SRL is applied in order to identify the arguments of the LF of the verb, how the constituents are related to the verb: the subject or any verb complement such as direct object, indirect object and so on.

This section is organized as follows: first, we give a general vision of the approach by a running example in order to clarify the complete process. Then, we describe with detail the use of rules and how we have obtained such rules. A description of the classifier is provided and finally we have included a comparison between our approach and some related works or topics.

4.1 Use case

This section shows an example of the global running of the proposed mixed method, using the resources described previously. The Spanish example is: '*Carlos lee los titulares deportivos de la Estrella Digital*' ('*Carlos reads the Estrella Digital sports headers*'). The expected LF is:

$$Carlos_N(x_1)\ leer_V(e_1,x_1,x_2)\ titular_N(x_2)\ deportivo_J(x_2)\ de_P(x_2, x_3)$$
$$Estrella_Digital_N(x_3)$$

Estrella_Digital is marked as noun because the dependency parser detects this expression as a named entity. Thus, the algorithm is applied as follows:

(1) The first step requires us to obtain the syntactic tree of the sentence (Figure 2).
(2) Then, each head word[3] of the sentence and its POS tag has an associated predicate. For example, $Carlos_N(x_1)$ or *titular*_N($x2$).
(3) For each noun phrase, intra-phrase transformation rules are applied at the same level. In this example, it is detected that *deportivo* is an adjective of *titular*, so they share the same argument (Table 3[4]).

[3] Informally, we might say that the head word is the main idea of the phrase.
[4] The POS tag of the rules are described according to those used in Penn Treebank. FreeLing uses a slightly different POS tag set. For example, adjectives are usually referred as a 'J' letter, whereas FreeLing uses a 'A' letter.

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Table 3. *Example of a rule applied at the same level*

Rule	Applied to	Result
NP→NJ	titular_N(x_2) deportivo_J(_)	titular_N(x_2) deportivo_J(x_2)
PP→P NP	de_P(_,_) Estrella_Digital_N(x_3)	de_P(_,x_3) Estrella_Digital_N(x_3)

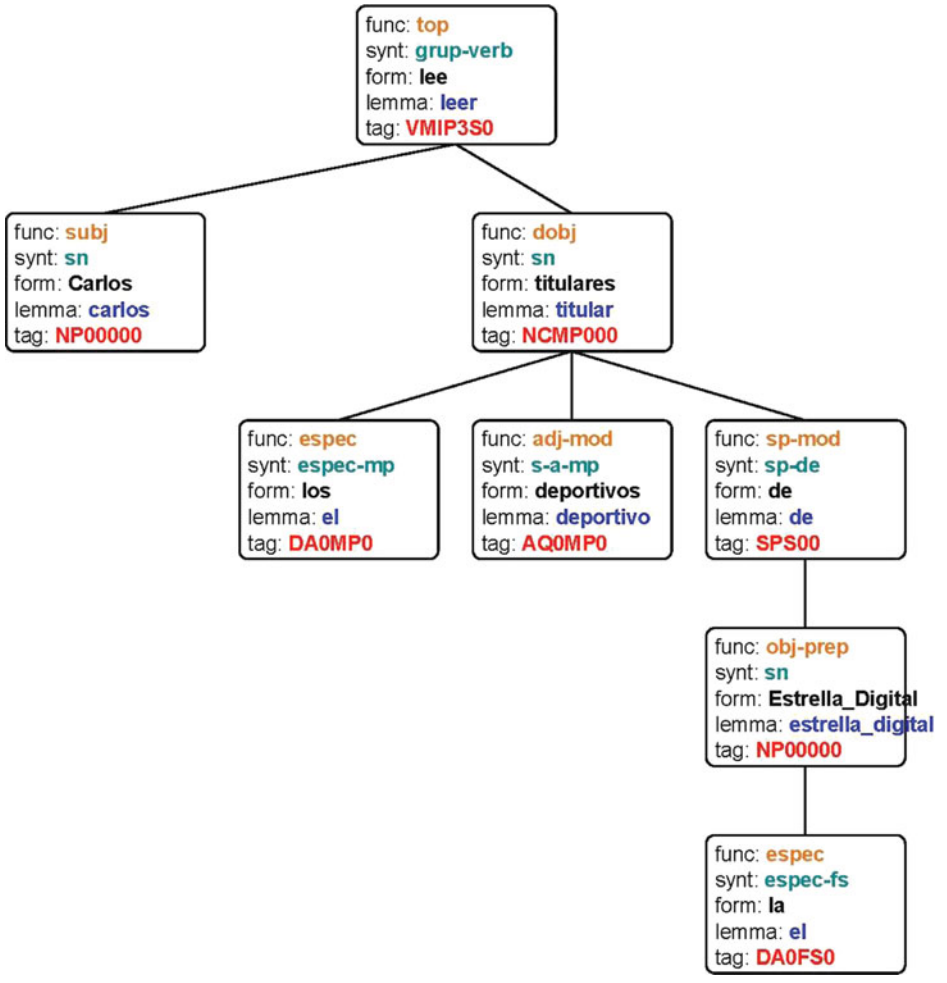


Fig. 2. (Colour online) Dependency tree of the phrase '*Carlos lee los titulares deportivos de la Estrella Digital*' ('*Carlos reads the Estrella Digital sports headers*') generated by FreeLing 3.1.

(4) The LF obtained are spread backwardly, and the intra-phrase transformation rules are applied recursively (Table 4).
(5) A feature vector is extracted for each noun phrase depending directly of the verb. An example of such a feature vector is depicted in Table 5.
(6) Finally, the classifier is applied for each feature vector obtained in the previous step in order to provide the arguments of the verb predicate.

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Table 4. *Example of a rule applied vertically*

rule	applied to	result
S→NP V NP PP	Carlos_N(x_1) leer_V(e_1,_,_,_,_) titular_N(x_2) deportivo_J(x_2) de_P(_,x_3) Estrella_Digital_N(x_3)	Carlos_N(x_1) leer_V(e_1,_,_,_,_) titular_N(x_2) deportivo_J(x_2) de_P(x_2,x_3) Estrella_Digital_N(x_3)

Table 5. *Example of a vector of features for the noun phrase* 'titulares deportivos de la Estrella Digital' *('Estrella Digital sports headers')*

Feature	value
Phrase type	NP
Main category	V
Syntactic tree path	NP,V
Position	right
Verb voice	active
Main word	titular
Joint of POS	NJPDNJ

At the end of steps 3 and 4, we have obtained the LF of non-verbal constituents and an incomplete LF of the verb predicate (it has no known arguments). Step 5 and 6 provide the arguments of the verb predicate in order to complete the LF of the sentence. A global view of the process is depicted in Figure 3.

4.2 LFi based on rules applied to non-verbal constituents

First, a set of hand-coded transformation rules are recursively used to derive the LF of each non-verbal constituent according to the type. For verbs, the LF is partially derived since the arguments of the corresponding predicate are unknown. Since, we are using transformation rules for non-verbal constituents we have only written what (Rus 2002) calls *intra-phrase transformation rules*. They generate predicates for

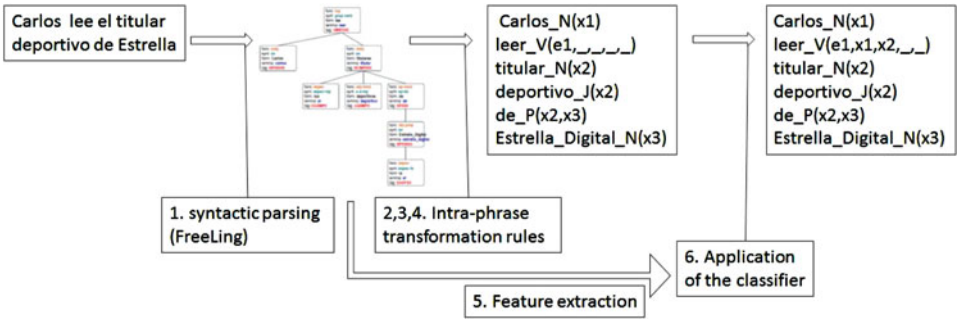


Fig. 3. (Colour online) Global view of the pipeline followed by the use case

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

every noun, verb, adjective or adverb. They also assign the variables that describe dependencies local to the phrase. The way that these intra-phrase transformation rules are obtained is depicted below. A set of candidates for verb arguments are obtained at the end of this recursive process. We use an approach based on rules because the syntax of the Spanish constituents is roughly similar to that of the English ones, and systems based on rules for English exceed 90% precision (Rus 2004b). In addition, the application of these rules is usually not an ambiguous task, and these constituents of phrases are relatively simple. Examples of such transformation rules are depicted in Tables 3 and 4.

4.2.1 Obtaining the transformation rules

The implementation of deriving LFs relies on information provided by the syntactic parser. We have developed a set of intra-phrase transformation rules that create predicates. We have written a transformation rule which produces the corresponding logical form for every selected rule of the parser. Some rules are rejected as we depict below.

(1) The Spanish corpus AnCora-ES (Taulé, Martí and Recasens 2008) is parsed obtaining a set of grammar rules. AnCora-ES is a Spanish corpus with syntactic, semantic and pragmatic annotations. AnCora is depicted in Section 5.
(2) Then the set of grammar rules is filtered:
    - Grammar rules relative to verb predicates are rejected because verb predicates are managed by using the machine learning module.
    - In order to obtain a reasonable number of grammar rules we have discarded grammar rules with 5 or fewer occurrences.
(3) At the end of the process, a transformation rule was written for each selected grammar rule.

In this way, we have obtained 73 rules adapted to Spanish syntax. For example, a valid Spanish grammar rule is NP →NJ (*'La casa blanca'*) so we create a transformation rule in order to obtain the LF of this phrase constituent. 94% of the non-verbal constituents are covered by using this set of transformation rules.

4.3 Derivation of the arguments of the verb predicate using a SRL classifier

Rus (2002) proposes the so-called *inter-phrase transformation rules* in order to provide the arguments of the verb predicates. Given the ambiguity of Spanish syntax we propose the replacement of such rules by using an approach similar to SRL: each constituent is classified as a semantic role, but instead of finding out the role, the system infers which is the argument with the logical form of the verb. The way that the system makes this inference is similar to the solution proposed by Gildea and Jurafsky (2002) for semantic role detection by means of machine learning algorithms. We apply this solution adapted to Spanish with the available resources, which are depicted in Section 5. Seven features are extracted from the constituent by using a dependency tree parser and they are used to train a classifier:

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

(1) **Phrase type** could be a noun phrase (NP), a prepositional phrase (PP), an adjective phrase($ADJP$) or an adverbial phrase($ADVP$).
(2) **Main category** has only two values, subjects (S), and verb phrases(VP), corresponding to subjects and objects of verbs, respectively.
(3) **Syntactic tree path**: determines the syntactic tree path from the constituent to the verb.
(4) **Position** of the constituent related to the verb (left or right).
(5) **Verb voice**: active or passive.
(6) **Main word** of the constituent. This word is marked by the *parser* according with some heuristic rules described in (Collins 1999). In Section 5 we describe the parser used.
(7) **The joint of the Part Of Speech (POS)** of each constituent word. For example, the phrase '*el año pasado*' takes the value $\{DNJ\}$.

The first six features are those used by Gildea and Jurafsky (2002). The POS joint is used in some SRL systems (Punyakanok *et al.* 2004; Carreras, Màrquez and Chrupała 2004b; Palmer *et al.* 2010).

4.4 Comparing the mixed method with similar studies

4.4.1 The mixed model and semantic role labeling

Although there is a similarity between the semantic role identification and the arguments of the logical form of the verb, there are important differences:

- LF uses 'slot-allocation' representation: the position of the arguments is fixed for the purpose of a simpler notation. Of course, it is possible to represent sentences without a direct object, for example. In such a case, the value of the 'direct object' argument of the normal form of the verb will be a null value, which cannot be referenced in any predicate. 'Null arguments' are not usually written if these are located at the end of the predicate.
- Since it is necessary to group different roles with the same argument in the logical form of the verb, the detection of these arguments is a simplification of the semantic role classification task.
- The detection of the limits of each semantic role in a phrase is a more difficult task than the classification. Nevertheless, the detection of the arguments of the logical form of the main verb can be achieved by applying the codified syntactic rules, so the semantic role classification task can be compared with the identification of the arguments of the logical form of the main verb.

In brief, the classifier of our proposal does not require semantic roles at all, it just learns the position of each argument of the LF of the verb. It is a simpler task than SRL and this is the reason why we apply a simplified version of SRL, as depicted in Section 4.3.

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

4.4.2 The mixed model and other LFi approaches using machine learning classifiers

The present study is similar to the approach presented in Rus (2004a). Rus proposes detecting functional arguments of verbs using decision trees. In a similar manner, the present study addresses the task of detecting logic roles as a classification problem. In addition, this problem is resolved using machine learning techniques and a corpus annotated with syntactic information useful for learning functional logic arguments. But there are some relevant differences: (i) since the language is different (English and Spanish) the resources are different; (ii) Rus (2004a) is interested in ambiguous verbs and sentences with two verbs in a coordination. For example, in the following sentence:

The judges hear or try a court case

there are two verbs, 'hear' and 'try' and one direct object, *'a court case'*. The issue is whether the direct object is shared by the verbs or not. But it seems clear that *'a court case'* is a direct object. This is not necessary true for other examples with a more ambiguous syntax. For example, the sentence *'La columna sostiene la casa'* (*'The column holds the house'*) is ambiguous since *'La columna' could be* subject or direct object. These kinds of examples are relatively frequent in Spanish, and are possible in other languages such as English. For this reason, one issue of the present study is whether a noun phrase is a subject, a direct object or something else.

5 Experiments and results

5.1 Experimentation framework

In our experiments, we have used the linguistic analysis software FreeLing 3.1 (Carreras *et al.* 2004a; Padró and Stanilovsky 2013), the semantic annotated corpus AnCora-ES (Taulé *et al.* 2008) and the machine learning software TimBL (Daelemans *et al.* 2004). The set of phrases and their LF, used to evaluate the complete system, were translated from the LFi task of Senseval-3 (Rus 2004b):

- FreeLing is a freely available linguistic analysis software package. Supported languages are Spanish, Catalan, French, Galician, Italian, English, Russian, Portuguese, Welsh and Asturian. It implements a *POS tagger*, a morphological analyzer and a rule-based dependency parser. The syntactic rules and the features extracted from each constituent are derived from the syntactic tree that *FreeLing* returns. Syntactic analyzers such as Collins (1999) provide the head (the nucleus of the phrase, not a complement). Since FreeLing provides superficial syntactic analysis, syntactic trees are flat, so the number of necessary syntactic rules is higher than in the case of deep analysis. FreeLing has already been used in several industrial projects (Padró and Stanilovsky 2013).
- AnCora-ES is a Spanish corpus with syntactic, semantic and pragmatic annotations. As well as the syntactic annotation, it carried out a semantic annotation by means of the synsets of the different wordnets built for this language, as well as an annotation of anaphoric and elliptic elements just as

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

the co-reference. The corpus extension is 100,000 words. The AnCora corpus also is available for Catalan, AnCora-CA is a corpus with similar coverage to AnCora-ES for Catalan. The annotated data is necessary in order for a machine learning system to learn which argument of the logical form of the verb is mapped to the given constituent. For this purpose, each syntactic function has an argument of the logical form of the associated verb, as we can see in Table 6. On the other hand,this syntactic information is provided by AnCora-ES but shallow parsers does not provide such kind of information. Note that our proposal is well suited to be applied by using just shallow parsing, if dependency or constituency parsing is no available.

- TiMBL is an open source software package implementing several learning algorithms, among them: IB1-IG, an implementation of k-nearest neighbor classification with feature weighting suitable for symbolic feature spaces, and IGTree, a decision-tree approximation of IB1-IG. TiMBL can work with alphanumeric features and performs multi-class classification, so it is useful for NLP tasks. Memory-based language processing is based on the idea that NLP problems can be solved by storing annotated examples of the problem in their literal form in memory, and applying similarity-based reasoning to these examples in order to solve new ones (Daelemans and van den Bosch 2005). For the experimentation we use the TiMBL 6.4 version. For the past decade, TiMBL has mostly been used in NLP as a machine learning classifier component including SRL for Spanish and Catalan (Morante and Busser 2007).

  We parameterized the algorithm empirically using the training testset: we found that IB1-IG obtains the best results. Information gain weighting was used to compute relevance. The distance metric is the cosine. We set the confidence threshold to 0.86 because we found that 90% of training examples were correctly classified and the coverage was 94.2% ($F_1 = 0.92$).

5.2 Experiments and results

In order to evaluate the experiments we considered four performance measures (Rus 2004b):

- Precision at **argument level** is the number of correctly identified arguments divided by the number of all identified arguments. Recall at argument level is the number of correctly identified arguments divided by the number of arguments that were supposed to be identified.
- Precision at **predicate level** is the number of correctly and fully identified predicates (with ALL arguments correctly identified) divided by the number of all attempted predicates. Recall at predicate level is the number of correctly and fully identified predicates (with ALL arguments correctly identified) divided by the number of all predicates that were supposed to be identified.

We evaluated both the classifier and the mixed method in several ways:

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Table 6. *Mapping between syntactic functions annotated in the AnCora-ES corpus and the corresponding argument position in the LF of the verb*

Syntactic function	Equivalent English function	Argument position
SUJ	subject	2
CD	direct object	3
CD.NF	direct object	3
CD.Q	direct object	3
CI	indirect object	4
CI.DATI	indirect object	4
CI.NF	indirect object	4
ATR	attribute	5
CPRED	predicative complement	5
CPRED.SUJ	predicative complement	5
CPRED.CD	predicative complement	5
CAG	agentive complement	5
CREG	prepositional complement	5

- The performance of identification of verb arguments using SRL, as depicted in the following section.
- The global performance at predicate and argument level.
- The performance of the identification of verb arguments using a rule-based approach instead of the proposed mixed method.
  As is described above, these rules are based on previous studies (Rus 2002, 2004b). Richer rules could be developed, for instance to improve the treatment of ambiguity, but this would be more complex work that we plan to perform in the near future.

5.2.1 Evaluation of the identification of verb arguments using the SRL module

Before evaluating the proposed LFi model, we show the results of the experiments conducted to verify that the SRL module functions properly. Our learning problem is that of classifying each constituent of the phrase attending to its grammatical relation with the verb: subject, direct object, indirect object and other syntactic functions. Thus, we extracted seven features for each constituent which are depicted in Section 4.3. An example of this feature vector is depicted in Table 5. The evaluation of the machine learning module in predicting the arguments of the logical form of the verb was carried out using 15,000 constituents for training and 1,000 for testing, extracted from AnCora-ES. Table 7 shows the classification of the 1,000 testing constituents. We obtained state-of-the-art results or even better values for recall (Morante and Busser 2007; Bick and Valverde 2009). The reason for this is that the test set was parsed manually. The following section builds up this hypothesis.

As we have shown, the average precision obtained was 84% for verb arguments at argument level, and a recall of 0.81 ($F_1 = 0.83$). The precision at predicate level was 89.2% of predicate correctly and fully identified. The recall in this case was 0.91 ($F_1 = 0.90$). The machine learning module achieved slightly better results at

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Table 7. *Confusion matrix of the 1,000 evaluated arguments (with sentences extracted from the AnCora-ES corpus). The position of the argument is codified according to Table 6*

	5	3	2	4
5	439	82	0	1
3	20	164	0	13
2	0	0	238	0
4	10	6	0	27
Precision	0.84	0.83	1	0.62
Recall	0.94	0.92	1	0.66
F1	0.89	0.87	1	0.64
Average precision: 0.84				

predicate level than at argument level. When a verb predicate is wrong, it usually has several errors in its arguments. These cases contain only one error at predicate level, but more than one error at argument level. We have found that predicates with arguments partially identified are not as usual as the whole arguments wrongly identified. This result is consistent with the ones reported in Rus (2004b).

For example, sentences which have both indirect and direct object have their positions sometimes wrongly swapped because the machine learning module fails. In such a case, the direct object is erroneously assigned to the fourth slot of the verb predicate, while the indirect object is assigned to the third slot. Then, the argument level measure is penalized with two errors while the predicate level measure is penalized with one error.

An interesting result is the exact precision achieved in the subject detection (category 2), 238 good choices out of 238. Despite the syntactic flexibility of the Spanish language, the subject detection is not a complex task because the subject usually appears at the left of the verb with a noun as headword. These two features were taken into account during the training.

The direct object (category 3) and the other predicative complements (category 5) obtained good results, between 85% and 90%. Nevertheless, the indirect object detection (category 4) obtains the poorest result, with a precision near to 65%. Almost all errors of the direct and indirect objects were caused by constituents wrongly marked as predicative complements. The main reason is that there are so many prepositional phrases, and the system was over-trained for category 5 (which contains all the verb complements that are not direct or indirect objects). As we can see in the confusion matrix, the great majority of the verb complements, almost 70%, belong to this fifth category.

In order to improve these results the system should use semantic information. One advantage of the use of machine learning algorithms instead of the ones based on rules is the integration of new knowledge sources. Systems based on rules can also incorporate such information, but they would need more complex rules that would need to be constructed manually. Without these knowledge sources the system cannot find the argument of a certain constituent. For instance, in the sentence *'Pedro*

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Table 8. *Some Spanish sentences and the logic form obtained*

Juan vuela desde Tokio hasta Nueva York (Juan flies from Tokio to New York)	Juan_[P](x_1) volar_[V](e_1 x_1) desde_[P](e_1 x_2) Tokio_[N](x_2) hasta_[P](e_1 x_3) Nueva_York_[N](x_3)
John es golpeado por una pelota (John is hit by a ball)	John_[N](x_1) golpear_[V](e_1 x_2 x_1) por_[P](e_1 x_2) pelota_[N](x_2)
En vez de alubias comería pizza (I'd eat pizza instead of beans)	En_vez_de_[P](x_2 x_1) alubias_[N](x_1) comer_[V](e_1 _ x_2) pizza_[N](x_2)
El baloncesto y el tenis son grandes deportes (basketball and tennis are great sports)	baloncesto_[N](x_1) y_[C](x_3 x_1 x_2) tenis_[N](x_2) ser_[V](e_1 x_3 x_4) grande_[A] (x_4) deporte_[N](x_4)
Un periodo de descanso permitiría el profesor (The teacher would allow a rest period)	periodo_[N](x_3) de_[P](x_3 x_2) descanso_[N](x_2) permitir_[V](e_1 x_1 x_3) profesor_[N](x_1)

recomienda a el niño que lea' the system should mark '*a el niño*' as indirect object. Nevertheless, in the sentence '*Ana quiere a el niño*' the same constituent appears with a similar syntactic tree, but '*a el niño*' should be marked as direct object. In order to resolve these situations it is necessary to include additional knowledge about the verb, the semantic roles that it can receive and the argument type. A resource such as FrameNet (Baker, Fillmore and Lowe 1998) could learn that '*lo querido*' is applied to people, and WordNet will provide the information that a boy is a person.

5.2.2 Evaluation of the mixed method

With the aim of testing the complete method, and constructing the rule base and the machine learning parts, we ran an experiment to obtain the LF of 150 sentences from the *Senseval-3 Logic Forms workshop*. These selected sentences were translated into Spanish manually. The LF of these sentences were adapted from English to Spanish. After the application of the hand-coded rules, we obtained 1,239 constituents distributed over these 150 sentences which must be classified as subject, direct object, indirect object or others. Table 8 shows some Spanish sentences and the LF obtained.

Table 9 shows the precision obtained with the machine learning algorithm, over these 1,239 verb arguments. The average precision is 75%. Note that the identification of verb arguments is better for AnCora-ES (see Table 7). This is not an unexpected result because the evaluation of the AnCora-ES was performed with

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Table 9. *Confusion matrix over the 1,239 verb arguments evaluated (150 translated sentences from the* Senseval-3 LFi *workshop). The position of the argument is codified according to Table 6*

	5	3	2	5
5	288	58	0	0
3	155	219	0	0
2	49	0	268	3
4	0	55	0	147
Precision	0.83	0.59	0.84	0.73
Recall	0.59	0.66	1	0.98
F1	0.89	0.87	1	0.64
Average precision: 0.75				

a manually tagged corpus, so there were no mistakes due to a wrong syntactic analysis.

The average precision at predicate level was 82.3% and the coverage 85.1% ($F_1 = 0.84$). This is a very similar result to the one obtained in Senseval-3 for the best groups for English (Rus 2004b).

Finally, we performed a straightforward analysis of errors at predicate level in order to detect the main sources of errors:

- The predicate of verbs are responsible for 62% of the errors. This explains why we obtain the best results when we take into account the whole of the predicates. Otherwise, this is an expected result because of the inherent difficulty in identifying verb arguments. The way to reduce this source of errors is to improve the SRL module.
- FreeLing is a high-precision parser but it is not perfect. 23% of errors have their origin in the parsing phase.
- Noun phrases have less ambiguity than verb phrases but there are significative cases of ambiguity in noun phrases. For example, coordinated compound nouns such as *'helado o flan casero'* (*'homemade ice cream or custard'*). The doubt here is the role of the adjective *'casero'*. It could refer to the noun *'flan'* only or in addition to *'helado'*. We have detected 6% of errors for this and other similar constructions.

5.2.3 Comparing the mixed and rule-based methods

In order to test the improvement of the application of machine learning we developed a version of the system based only on rules. We replaced the machine learning module by using 19 inter-phrase transformation rules (Table 11). We applied the same approach used to obtain the intra-phrase transformation rules in order to obtain the most frequent grammar constructions (Section 2.1). Our system based on rules is made up by 19 inter-phrase transformation rules plus 73 intra-phrase transformation rules, 92 rules. Thus, this 92 rules are our version of the 70 rules proposed in Rus (2002) to cover all WordNet glosses adapted to Spanish.

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Table 10. *Examples of transformation rules for verb phrases*

Grammar rule	Transformation Rule (LFT)	Phrase
VP → VP PP	verb/VP-PASS por/PP(_,x) → verb(e,x,_) & por(e,x)	(VP (querido/VBN por/PP))
VP → VBZ NP	verb/VBZ noun/NP(x) → verb(e,_,x) & noum(x)	(VP (quiere/VBZ pan/NP))
VP → VB ADVP	verb/VBZ con/PP(x) → verb(e,_,_,_,x)	(VP (viaja/VBZ con/PP))

Table 11. *The 19 grammar rules for verb phrases*

VP → VP PP	VP → VBZ	VP → VB ADVP
VP → VBZ NP	VP → NP VBZ	VP → VBN S
VP → VBZ NP PP	VP → VBZ PP NP	VP → VBZ ADJP NP
VP → VBZ NP NP	VP → ADVP VB	VP → VB PP
VP → PP PP VB	VP → VBG PP PP PP	VP → VB NP ADVP
VP → VP ADVP NP	VP → ADVP VB	VP → ADVP VB
VP → PP VP		

Table 10 shows examples of such rules. Note that these 19 rules are not required for the mixed method because the SRL module replaces such rules.

Table 12 shows that the proposed mixed method outperforms the rules-only based system by 11%. The improvement is greater in the classification of subject (category 2) and direct object constituents (category 3), just where word order in Spanish is more variable. For instance, the mixed method obtains the correct LF of the phrase '*Un periodo de descanso permitiría el profesor*' ('*the teacher would allow a rest period*'). However, the rules-only based system makes a mistake confusing subject and direct object. By contrast, the mixed method takes into account the 'Noun phrase type'

Table 12. *Confusion matrix over the 1,239 arguments comparing both models. The position of the argument is codified according to Table 6*

	5	3	2	4	5	3	2	4
5	288	58	0	0	288	57	0	1
3	155	219	0	0	184	190	0	0
2	49	0	268	3	135	0	181	4
4	0	55	0	147	0	55	0	147
Precision	0.83	0.59	0.84	0.73	0.83	0.51	0.56	0.73
Recall	0.59	0.66	1	0.98	0.47	0.63	1	0.97
F1	0.89	0.87	1	0.64	0.6	0.56	0.72	0.83
	Average precision: 0.74				Average precision: 0.67			
	Mixed model (with machine learning module)				Replacing machine learning with 19 hand-coded rules			

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

feature (and others) and it helps the machine learning module to correctly identify *'El profesor'* as the subject and *'un periodo de descanso'* as a direct object.

By contrast, both rule-based and mixed methods give the same performance when detecting adverbial constituents (category 4). Syntatic rules work well because the prepositions of such constituents identify the position of the corresponding argument in the verb predicate.

6 Conclusions and future work

This paper presents a new method of deriving LF, based on rules and machine learning, from the AnCora-ES corpus. This method was applied to Spanish, a complex language due to the flexibility of its syntax. For this reason, a rule-based system cannot obtain results as good as with other languages like English. Nevertheless, the use of machine learning allows for the incorporation of varied knowledge. In this study, the system uses machine learning as described by Gildea and Jurafsky (2002), to classify semantic roles. The experiments performed on the Senseval-3 and Senseval-3 LFi data demonstrate the consistency of our approach. Similarly, the next step will be the evaluation of the proposed method over other Romance languages such as Catalan, French and Italian, and other sets of sentences and LF. In addition, a general improvement could be achieved in a straightforward manner by improving the performance of two key elements: the machine learning module and the dependency trees:

- In order to improve the machine learning module we could adapt more sophisticated features taking advantage of FrameNet such as those proposed in Màrquez *et al.* (2008) and Palmer *et al.* (2010). For example, we could use various lexical and syntactic features such as syntactic n-grams, phrase type and grammatical function calculated over the annotated corpora of Spanish FrameNet.
- FreeLing performs well in obtaining the dependency tree for Spanish but this could not be necessarily true for languages other than Spanish. A voting system using state-of-art parsers (Henderson *et al.* 2013; Johansson and Nugues 2008; Zhao *et al.* 2009) could be a good way to improve the system, to decrease the number of rules and to apply the model to other languages than Spanish.

References

Agerri, R., and Peñas, A. 2010. On the automatic generation of intermediate logic forms for WordNet glosses. In A. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing*, pp. 26–37. Lecture Notes in Computer Science, vol. 6008. Berlin Heidelberg: Springer.

Ahn, D., Fissaha, S., Jijkoun, V., and De Rijke, M. 2004. The University of Amsterdam at Senseval-3: semantic roles and logic forms. In R. Mihalcea, and P. Edmonds (eds.), *Senseval-3: 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain: Association for Computational Linguistics, pp. 49–53.

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Alfaraz, G. 2012. Word order as a change in progress: evidence from Cuban Spanish. In *Proceedings of the 6th International Workshop on Spanish Sociolinguistics*. University of Arizona, published by Cascadilla Proceedings Project, Somerville, MA, USA.

Anthony, S., and Patrick, J. 2004. Dependency based logical form transformations. In *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, pp. 54–57.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. 1998. The Berkeley FrameNet project. In *Proceedings of the 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, Université de Montreal, Canada, pp. 86–90.

Bick, E., and Valverde, M. 2009. Automatic semantic role annotation for Spanish. In *Proceedings of NODALIDA*, Odense, Denmark, pp. 215–218.

Carreras, X., Chao, I., Padró, L., and Padró, M. 2004a. FreeLing: an open-source suite of language analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal.

Carreras, X., Màrquez, L., and Chrupała, G. 2004b. Hierarchical recognition of propositional arguments with perceptrons. In *Proceeding of CoNLL'2004 Shared Task: Semantic Role Labeling*. Boston, MA, USA.

Collins, M. J. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, Philadelphia: University of Pennsylvania.

Daelemans, W., and van den Bosch, A. 2005. *Memory-Based Language Processing*, Cambridge: Cambridge University Press.

Daelemans, W., Zavrel, J., van der Sloot, K., and van den Bosch, A. 2004. *TiMBL: Tilburg Memory-Based Learner, version 5.1, Reference Guide*. ILK Technical Report 04-02.

Delmonte, R., and Rotondi, A. 2012. Treebanks of logical forms: they are useful only if consistent. In *LREC 2012, ISA7 Workshop*.

Ferrández, Ó., Terol, R. M., Muñoz, R., Martínez-Barco, P., and Palomar, M. 2007. A knowledge-based textual entailment approach applied to the AVE task. In C. Peters, P. Clough, F. C. Gey, J. Karlgren, B. Magnini, D. W. Oard, M. Rijke, and M. Stempfhuber (eds.), *Evaluation of Multilingual and Multi-modal Information Retrieval*, pp. 490–493. Lecture Notes in Computer Science, vol. 4730. Berlin Heidelberg: Springer.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., Schlaefer, N., and Welty, C. 2010. Building Watson: an overview of the DeepQA project. *AI Magazine* **31**(3): 59–79.

Fowler, A., Hauser, B., Hodges, D., Niles, I., Novischi, A., and Stephan, J. 2005. Applying COGEX to recognize textual entailment. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*. Southampton, U.K.: PASCAL Recognising Textual Entailment Challenge, pp. 69–72.

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

Harabagiu, S. M., Miller, G. A., and Moldovan, D. I. 1999. Wordnet 2-A morphologically and semantically enhanced resource. In *Proceedings of SIGLEX*, Vol. 99. College Park, Maryland, USA.

Henderson, J., Merlo, P., Titov, I., and Musillo, G. 2013. Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational linguistics*, **39**(4), 949–998.

Johansson, R., and Nugues, P. 2008. Dependency-based semantic role labeling of PropBank. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Waikiki, Honolulu, Hawai, USA, pp. 68–791.

Lepore, E., and Ludwig, K. 2001. What is logical form?. In P. Kotatko, P. Pagin, and G. Segal (eds.), *Interpreting Davidson*, pp. 111–142. Stanford: CSLI, 2001.

Màrquez, L.,Carreras, X., Litkowski, K., and Stevenson, S. 2008. Semantic role labeling: an introduction to the special issue. *Computational Linguistics* **34**(2): 145–159.

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

McCord, M. C., Murdock, J. W., and Boguraev, B. K. 2012. Deep parsing in Watson. *IBM Journal of Research and Development* **56**(3.4): Berlin: Springer.

McCune, W. W. 1994. OTTER reference manual and guide. *Argonne National Laboratory, Illinois*.

Moldovan, D., Clark, C., Harabagiu, S., and Hodges, D. 2007. Cogex: a semantically and contextually enriched logic prover for question answering. *Journal of Applied Logic* **5**(1): 49–69.

Moldovan, D. I., and Rus, V. 2001. Logic form transformation of WordNet and its applicability to question answering. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. ACL '01, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 402–409.

Morante, R., and Busser, B. 2007. Memory-based semantic role labelling of Catalan and Spanish. In *Proceedings of Conference on Recent Adavances in Natural Language Processing RANLP-2007*, Borovets, pp. 388–394.

Muñoz-Terol, R., Martínez-Barco, P., and Palomar, M. 2007. Applying logic forms and statistical methods to CL-SR performance. In *Evaluation of Multilingual and Multi-modal Information Retrieval*, pp. 766–769. Lecture Notes in Computer Science, vol. 4730. Berlin: Springer.

Nakamura, M., Kimura, Y., Pham, M. Q. N., Nguyen, M. L., and Shimazu A. 2010. Treatment of legal sentences including itemization written in Japanese, english and Vietnamese towards translation into logical forms. *Journal Natural Language Processing* 81–100.

Nguyen, M. L., and Shimazu, A. 2014. A semi supervised learning for mapping NL sentences to logical form with ambiguous supervision. In *Data and Knowledge Engineering*.

Padró, L., and Stanilovsky, E. 2013. FreeLing 3.0: towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012). ELRA*, Istanbul, Turkey.

Palmer, M., Gildea, D., and Xue, N. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies* **3**(1), 1–103.

Pietroski, P. 2009. Logical form. *The Stanford Encyclopedia of Philosophy* (Fall 2009 Edition), E. N. Zalta (ed.). http://plato.stanford.edu/entries/logical-form/

Punyakanok, V., Roth, D., Yih, W.-tau, and Zimak, D. 2004. Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th International Conference on Computational Linguistics*. COLING '04, Stroudsburg, PA, USA: Association for Computational Linguistics, p. 1346.

Rus, V. 2002. *Logic Form For WordNet Glosses and Application to Question Answering*. Ph.D. thesis, Computer Science Department, School of Engineering, Southern Methodist University, Dallas, Texas.

Rus, V. 2004a. Experiments with machine learning for logic arguments identification. In *Proceedings of the 15th Midwest Artificial Intelligence and Cognitive Science Conference MAICS 2004*. Chicago: Omnipress, pp. 40–47.

Rus, V. 2004b. A first evaluation of logic form identification systems. In *Proceedings of Senseval-3: Third International Workshop on Evaluation of Systems for Semnatic Analysis for Text*, Barcelona, Spain: Association for Computational Linguistics, pp. 37–40.

Russell, B. 1914. *Our Knowledge of the External World: As a Field for Scientific Method in Philosophy*, p. 53. New York: Routledge.

Suñer, M. 1982. *Syntax and Semantics of Spanish Presentational Sentence-types*, Romance languages and linguistics series. Georgetown: Georgetown University Press Washington, DC.

Tatu, M., Iles, B., and Moldovan, D. 2007. Automatic answer validation using COGEX. In *Evaluation of Multilingual and Multi-modal Information Retrieval*, pp. 494–501. Lecture Notes in Computer Science. Berlin: Springer.

Taulé, M., Martí, M., and Recasens M. 2008. AnCora: multilevel annotated corpora for Catalan and Spanish LREC.

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press

Todorova, Y. 2009. Answering questions from natural language using A-Prolog. In *Logic Programming*, pp. 544–546. Lecture Notes in Computer Science, vol. 5649. Berlin: Springer Berlin Heidelberg.

Tustison, C. A. 2004. *Logical form Identification for Medical Clinical Trials*. Ph.D. thesis.

Wenner, C. 2007. Rule-based logical forms extraction. In *Proceedings of the 16th Nordic Conference of Computational Linguistics NODALIDA-2007*. Tartu, Estonia: Tartu University, pp. 402–409.

Zhao, H., Chen, W., Kity, C., and Zhou, G. 2009. Multilingual dependency learning: a huge feature engineering method to semantic dependency parsing. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*. Boulder, Colorado: Association for Computational Linguistics, pp. 55–60.

https://doi.org/10.1017/S1351324915000297 Published online by Cambridge University Press