First steps on Automatic Semantic Role Labeling for Basque Verbs

Author: Haritz Salaverri Izco

Advisors: Olatz Arregi & Beñat Zapirain

hap

Master on Analysis and Processing of Language
Final Thesis

September 2013

Departments: Computer Systems and Languages, Computational Architectures and Technologies, Computational Science and Artificial Intelligence, Basque Philology, Electronics and Telecommunication.

Laburpena

Rol semantikoen etiketatzea (*SRL*) garrantzia handia hartzen ari den hizkuntzalaritza konputazionalaren barneko alorra da. Izan ere, *Association for Computational Linguistics* (*ACL*) erakundeak oinarrizko NLP (*Natural Language Processing*) ikerketa lerrotako dauka. Lengoaia Naturalaren Prozesamenduaren alorrean kokatzen diren aplikazioen garapena aurrera eraman ahal izateko, askotan, beharrezkoa izango da informazio semantikoa, eta hain zuzen ere, rol semantikoak etiketatuta dauzkaten corpusak eskura izatea. Makina bidezko itzulpenean eta testu-laburpen automatikoan, esaterako, lagungarria izan daiteke rolek eskaintzen duten informazioa, besteak beste emaitza hobeak lortu ahal izateko.

Lan honek euskarazko aditzen rol semantikoak era automatikoan etiketatzeko gaitasuna izango duen sistema baten garapen prozesua deskribatzen du. Honetan, *VerbNet/PropBank* eredua jarraitzen duen *EPEC-ROLSEM* corpusa erabili da. Hasiera batean, bi hurbilpen ezberdin aztertzen dira: Lehen hurbilpena erregela linguistikoetan oinarrituta rol semantikoak etiketatzen dituen sisteman datza. Bigarrena, ordea, ikasketa automatikoko teknikak (*Machine Learning*) erabilita garatutako sistema bat inplementatzean datza. Ondoren, azken etiketatzailea garatu da hurbilpen bakoitza inplementatzen duten sistemen gainean egindako esperimentuetatik lortu diren emaitzetan oinarrituta.

Abstract

Semantic Role Labeling (*SRL*) is a research area on the rise in the field of Natural Language Processing and is listed as a core *NLP* task by the *Association for Computational Linguistics* (*ACL*). As a matter of fact, having a large corpus with annotation of semantic roles is crucial for the development of applications and advanced systems for machine translation, language learning, text summarization and many others.

This paper describes the process followed to develop a system for the automatic labeling of semantic roles for Basque verbs. *EPEC-ROLSEM*, which is a corpus labeled at predicate level following the *VerbNet/PropBank* model, has been used for this purpose. At first, two different approaches are considered: The first one consists of a system that will label semantic roles based on a set of linguistic rules; a second approach consisting of a system developed using Machine Learning (ML) techniques is considered. Afterwards, the final tagger is implemented based on the results that were obtained from experiments, which had been performed on systems that used both approaches.

Contents

List of Figures

List of Tables

1 Introduction

This work is aimed at developing a system for the automatic labeling of semantic roles for Basque verbs. The system will be capable of labeling corpora on a large scale, drastically reducing the temporal and economic cost of annotating corpora manually. As previously stated, the resource of semantically labeled corpora is crucial for the development of applications and advanced systems in computational linguistics; therefore, it can be concluded that the system developed will fulfill the need for annotation of semantic roles within large Basque corpora (*EPEC-ROLSEM*).

In this section an introduction to the work is made. The section is divided into three subsections: the first one explains what semantic roles are, what the semantic role labeling task consists of, and what this can be useful for (subsection 1.1); a second subsection (1.2) explains what the linguistic theories regarding roles are and lists the computational resources available to develop semantic role taggers. Finally, the third subsection (1.3) describes the structure of the paper.

1.1 Semantic role labeling

An **event** is typically referred to as a fact or a *something that happens* in reality. For example, the **sentence** *Mike eats an apple* describes the event consisting of *an apple* being *eaten* by *Mike*. In natural language, facts or events are represented using sentences that can then be analyzed in different levels such as the syntactic and semantic levels. One event can have different representations in the human languages, that is, different sentences can refer to the same event, and sometimes, can have the same **meaning**. For example,

Sentence 1: *Mary drove a red car around the block.*

Sentence 2: *A red car was driven around the block by Mary.*

Both sentences refer to a same event where *a red car* has been *driven around the block* by *Mary*. Nevertheless, not all the sentences that refer to the same event always have the same meaning. For example,

Sentence 3: *Joe sold a red car to Mary for $5,000.*

Sentence 4: *Mary bought a red car from Joe for $5,000.*

In this example both sentences describe the same event but the meaning is different. Performing sentence-level semantic analysis can help determine *who* did *what* to *whom*, *where*, *when*, and *how* within an event. As it is stated in (Màrquez et al., 2008) the ***predicate*** of a clause (typically a verb) establishes *what* took place, and other sentence constituents express the participants in the event (such as *who* and *where*), as well as further event properties (such as *when* and *how*).

The main task of semantic role labeling (*SRL*), sometimes also called *shallow semantic parsing*, is to detect the **semantic relations** that hold among the predicate (verb) of a sentence and its associated ***participants*** and ***properties*** and the classification into their specific **roles**.

Semantic relations were introduced in generative grammar during the mid-1960s and early 1970s as a means of classifying the arguments of natural language predicates into a close set of participant types (roles), which were thought to have a special status in grammar. Semantic roles are also known as thematic roles, semantic cases, theta-roles (generative grammar), and deep cases (case grammar). Given the next sentence:

$$[Joe]_{Agent}[drank]_{(PREDICATE)}[a\ bottle\ of\ watter]_{Patient}.$$

Joe and *a bottle of watter* are the participants associated with the predicate *drank* (drink) and are classified into semantic roles as *Agent* and *Patient* respectively. Normally, the *Agent* role is assigned to event participants (subjects) that perform the action described in the predicate, in this case *drinking*. The *Patient* role instead is assigned to event participants (objects) that are unintendedly affected by the action, by the *what*, performed in the event.

Although there is no consensus on a definitive list of semantic roles among linguists some basic roles such as *Agent*, *Experiencer* or *Patient* are often considered in *SRL* for the entities participating in an event (participants), known as ***arguments***, and *Location*, *Temporal* and *Manner* for the characterization of other aspects of the event or participant relations (properties), known as ***adjuncts***. Given the next sentence:

$$[Yesterday]_{Temp}[Joe]_{Agent}[drank]_{(PRED)}[a\ bottle\ of\ watter]_{Patient}[in\ the\ porch]_{Loc}.$$

Time and place properties for the event are expressed by the *Temporal* adjunct *Yesterday* and the *Location* adjunct *in the porch*. The participants in the event are expressed by the *Agent* argument *Joe* and the *Patient* argument *a bottle of watter*.

Relevance of *SRL* to *NLP* applications

Roles represent a robust semantic relation between a predicate and its arguments that can be usefully exploited in *NLP* applications such as:

- ***Question Answering*** (*QA*) systems (Shen and Lapata, 2007). For example, faced with the question *What year did the U.S. buy Alaska?* and the retrieved sentence *...before Russia sold Alaska to the United States in 1867...*, a hypothetical *QA* system must identify that *United States* is the *Buyer/Agent* despite the fact that it is attested in one instance as a subject and in another as an object. Once this information is known, isolating the correct answer (i.e. 1867) can be relatively straightforward.
- ***Machine Translation*** (*MT*) systems. The information provided by semantic roles can be taken into account in order to correctly translate the arguments in a sentence

from one language to another. The meaning can change depending on the roles the arguments have in a sentence (Boas, 2002).

- ***Text summarization*** systems (Melli et al., 2005). Semantic roles can improve results in automatic text summarization.

In addition to the mentioned applications, semantic roles can also be useful for many other *NLP* tasks, like, for example, Information Extraction (*IE*), textual entailment systems, language learning and many others.

1.2 Semantic roles: Approaches and computational resources

Approaches (Linguistic theories)

There exist two main approaches toward semantic roles in linguistic theory. The first approach or linguistic theory focuses on explaining the varied expression of verb arguments within syntactic positions. The foundational work for this theory *English verb classes and alternations: A preliminary investigation* (Levin, 1993) concludes that the patterns of syntactic alternation exhibit a regularity that reflects an underlying semantic similarity among verbs, forming the basis for Levin verb classes. As it is stated in (Màrquez et al., 2008) such classes and the argument structure specifications for them, have proven useful in a number of *NLP* tasks, including *SRL* (Swier and Stevenson, 2004), and have provided the foundation for the computational verb lexicon ***VerbNet*** (Kipper et al., 2000).

The second approach, on the other hand, a theory of meaning called **frame semantics**, focuses on the idea that a word activates a **frame** of semantic knowledge that relates linguistic semantics to encyclopedic knowledge. The *frame* concept can help reveal that the sentences in the next example describe the same situation (event) but from different perspectives.

Sentence 1: *Joe sold a red car to Mary for $5,000.*

Sentence 2: *Mary bought a red car from Joe for $5,000.*

The foundational work for this theory is *Frame semantics and the nature of language* (Fillmore et al., 2004). As it is also noted in (Màrquez et al., 2008) the idea of a word activating a frame of semantic knowledge has tended to focus on the delineation of situation-specific frames (e.g., a *Commerce_sell* frame) and correspondingly more specific semantic roles (e.g., *Buyer*, *Goods* and *Seller*) that codify the conceptual structure associated with lexical items (Fillmore et al., 2004). This linguistic theory has provided the foundation for the lexical database of English ***FrameNet***.

VerbNet (VN)

VerbNet is one of the largest domain-independent **lexicons of English verbs** available nowadays on-line. It is structured in a hierarchical way and links the information contained in it with other publicly available linguistic resources such as the lexical databases of English *WordNet* and *FrameNet*. As it is stated in (Kipper, 2005) *VerbNet* has been created with explicitly stated syntactic and semantic information, using **Levin verb classes** to systematically construct lexical entries. Each verb class in *VN* is completely described by thematic roles, selectional restrictions on the arguments, and frames consisting of a syntactic description and semantic predicates with a temporal function. An example of a frame entry in the verb class *Hit-18.1.* from the lexicon is shown in figure 1.

Class Hit-18.1			
Roles and Restrictions: Agent[+int_control] Patient[+concrete] Instrument[+concrete]			
Members: bang, bash, hit, kick, ...			
Frames:			
Name	Example	Syntax	Semantics
Basic Transitive	Paula hit the ball	Agent V Patient	cause(Agent, E)manner(during(E), directedmotion, Agent) !contact(during(E), Agent, Patient) manner(end(E),forceful, Agent) contact(end(E), Agent, Patient)

Figure 1: Frame entry in Levin verb class from the *VerbNet* lexicon[1]

The example first shows the semantic roles that the predicate *hit* takes when being used with the sense described in the *VN* class (first sense: *18.1*), in addition, the **selectional restrictions** associated with each role are listed. Then, the **members** are shown; members are synonyms of the predicate in the class (in this case *hit-18.1*). This means that for example a proposition with the predicate *bang* behaves like predicate *hit-18.1* and for this reason it has the same features as the ones shown in class *hit-18.1*. Finally, a list of frames is given. Each frame consists of a specific syntactic and semantic structure. For example, the frame that corresponds to the proposition *Paula hit the ball* will have an *Agent_ Verb_ Patient* syntactic structure and the frame that corresponds to *Paula hit the ball with a stick* will have *Agent_ Verb_ Patient_ Instrument*.

As it has been previously stated, Levin verb classes and the argument structure specifications for them have provided the foundation for *VerbNet*. The **argument structure** of a verb, sometimes also called the predicate-argument structure, is the lexical information about the arguments of a predicate and their semantic and syntactic properties.

[1] http://verbs.colorado.edu/ mpalmer/projects/verbnet.html

```
(i)     John opened Bill's door (with his key)
        John's key opened Bill's door
        Bill's door opened
        Bill's door was opened (by John)
(ii)  a:  OPEN  (John   door    key)
                  |      |       |
                Agent   Theme   Instrument
      b:  OPEN  <Ag, Th, Instr>
```

Figure 2: Argument structure for the verb *Open*[2]

In the above example, the verb *Open* has an argument structure which induces obligatorily one argument position (Theme), and optionally two more (Agent and Instrument). This argument structure explains what the sentences in (i) have in common. The argument structure of *Open* can be indicated as in (ii) a or b.

PropBank (PB)

PropBank is a corpus that contains one million words from the *Wall Street Journal*. The text in the corpus is annotated with predicate-argument structures for verbs using semantic role labels for each verb argument. The argument structure specification used in this work is the same as the one used in the *PropBank* project. This specification is indicated in (Carreras and Màrquez, 2005). According to this, the arguments are classified into ***numbered arguments*** and ***adjuncts***. Numbered arguments (from *arg0* to *arg3*) define predicate-specific roles and their semantics depends on the predicate (verb) and the predicate usage in a sentence, or verb sense. Usually, *arg0* stands for the *Agent* and *arg1* corresponds to the *Patient* or *Theme* of the proposition. Adjuncts on the other hand define general arguments that any verb may take optionally, sometimes also called **adjunct-like roles**. Adjuncts are marked using the tag *argM* combined with a secondary tag indicating the type. Eleven different types have been considered in this study.

*argM*LOC*	Location
*argM*TMP*	Temporal
*argM*MNR*	Manner
*argM*CAU*	Cause
*argM*ADV*	General purpose
*argM*PRP*	Purpose
argM-*	Unknown type marker
*argM*NEG*	Negation marker
*argM*DIS*	Discourse marker
*argM*DIR*	Direction
*argM*MOD*	Modal

Table 1: Types of adjunct-like roles in predicate-argument structures

[2]http://www.glottopedia.org/index.php/Argument_structure

Given the sentence from the beginning, an example of the argument structure specification used in this work (and in *PropBank*) is shown:

$$Mary\ drove\ a\ red\ car\ around\ the\ block.$$

The semantic roles for the arguments are labeled like this:

$$[Mary]_{Agent}[drove]_{(PRED)}[a\ red\ car]_{Patient}[around\ the\ block]_{Location}$$

And the argument structure for the sentence is represented this way:

$$[Mary]_{arg0}[drove]_{(PRED)}[a\ red\ car]_{arg1}[around\ the\ block]_{argm*LOC}$$
$$(arg0\_PRED\_arg1\_argm*LOC)$$

The *PropBank* project (Kingsbury and Palmer, 2003) has been crucial in the research of natural language processing, and more precisely, in the semantic role labeling task. In fact, the development of such a corpus in addition with *VN* were the resources that led to the first system that labeled semantic roles using machine learning techniques. According to (Palmer, 2009) the primary goal of *PropBank* is to provide consistent, general purpose labeling of semantic roles for a large quantity of coherent text that can provide training data for supervised machine learning algorithms. An example from *PropBank* that shows a proposition with the verb *hit* is shown in figure 3.

```
<roleset id="hit.01" name="strike" vncls="18.1 18.4">
<roles>
  <role descr="agent, hitter - animate only!" n="0">
        <vnrole vncls="18.1" vntheta="Agent"/></role>
  <role descr="thing hit" n="1">
        <vnrole vncls="18.4" vntheta="Theme"/>
        <vnrole vncls="18.1" vntheta="Patient"/></role>
  <role descr="instrument, thing hit by or with" n="2">
        <vnrole vncls="18.4" vntheta="Location"/>
        <vnrole vncls="18.1" vntheta="Instrument"/></role>
</roles>

<example>
  <text>
    Bank of New England has been hit hard by the region's real-estate slump.
  </text>
  <arg n="1">Bank of New England</arg>
  <rel>hit</rel>
  <arg f="MNR" n="m">hard</arg>
  <arg n="2">by the region's real-estate slump</arg>
</example>
```

Figure 3: *PropBank* example for roleset *hit.01* (file *hit.xml* in *PB 1.7*)

The figure above shows the first roleset contained in file *hit.xml* from *PB* version 1.7. Each roleset is mapped with the corresponding *VN* classes, in this case *hit-18.1* from figure 1 and *hit-18.4*. A **roleset** indicates what semantic roles the examples contained in it will have. These roles depend on the *VN* classes to which the roleset is mapped. Figure 3 also shows an example contained in roleset *hit.01*.

FrameNet (FN)

FrameNet is a lexical database of English that follows the *frame semantics* paradigm previously presented. It contains around 1,200 semantic frames and 13,000 **lexical units** (*LU*). Lexical units are words linked to meanings. For example, the words *buy*, *sell* and *spend* evoke the *Commerce_sell* frame. If a word has multiple meanings (polysemous word) it is represented by several lexical units. In addition to these, the database also contains more than 170,000 manually annotated sentences that provide a unique training dataset for semantic role labeling[3].

Each frame in *FrameNet* is assigned a number of frame-specific semantic roles which are called **frame elements**. These elements are classified into **core** elements (numbered arguments in *PB*) and **non-core** elements (adjuncts in *PB*) as it is shown in the example below.

$$[Joe]_{Seller(C)}[sold]_{(PREDICATE)}[a\ red\ car]_{Goods(C)}[to\ Mary]_{Buyer(C)}[for\ \$5{,}000]_{Money(NC)}.$$

This example activates the *Commerce_sell* frame which consists of the *Seller*, the *Buyer* and the *Goods* core elements fulfilled in the example with *Joe*, *a red car* and *to Mary* accordingly; and several non-core elements such as the *Manner*, the *Means*, the *Money* (fulfilled with *for $5,000*) etc.

PropBank/FrameNet *example*

The next example shows the sentence, *Yesterday, Joe saw a blue bird* , to which semantic role labeling has been performed following both the *PropBank* and the *FrameNet* methods.

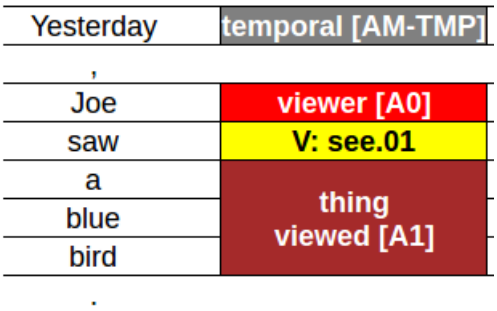

Figure 4: Semantic roles in *PropBank* and *FrameNet*[4]

[3]https://framenet.icsi.berkeley.edu/fndrupal/about

As it can be noticed in the figure, the arguments identified in the example-sentence are three: *Yesterday*, *Joe* and *a blue bird*. In *PropBank*, *Yesterday* is labeled with an *AM-TMP* mark indicating the argument is a temporal type adjunct, *Joe* is labeled with an *A0* mark indicating the argument is the subject (*Agent*) and finally *a blue bird* has the *A1* mark that indicates it is the object (*Patient*) of the sentence. In *FrameNet* on the other hand, the *see.01* frame is activated due to the presence of the predicate *saw*. The arguments in the sentence are labeled as the *Viewer* (*Joe*) and the *thing viewed* (*a blue bird*) core elements and the temporal non-core frame element *Yesterday*.
par

1.3 Paper structure

The article is divided into four sections. The first one is the *Introduction* (section 1); in it, what the semantic role labeling task consists of is treated first (subsection 1.1), then, a review is made of the linguistic approaches and the computational resources available to train *SRL* systems (subsection 1.2). The title for the second section is *State of the art* (section 2). In this section, a short explanation is made of the architecture that role labeling applications have nowadays. In addition, the structure that the *EPEC-ROLSEM* corpus used in this work has is presented in subsection 2.1. The third section is called *Experimentation and results* (section 3). In this section, the two approaches given to the labeling task (subsections 3.1 and 3.2) are treated by describing the experiments performed (subsections 3.1.1 and 3.2.1) and the results obtained (subsections 3.1.3 and 3.2.3), as well as the description of the system developed (subsections 3.1.2 and 3.2.2). Then, the development for the final *SRL* system that will use the approach that has turned out to be the best is treated in subsection 3.3. Finally, the fourth section is titled *Conclusions and future works* (section 4). In it, the conclusions drawn from developing the work are explained, and the tasks that could be performed in the future in addition to the problems encountered during the development process are discussed.

[4] http://cogcomp.cs.illinois.edu/demo/srl/results.php

2 State of the art

The history of computational linguistics is strictly connected to machine learning; in fact, the new algorithms and learning techniques developed in the past two decades have made it possible for the *NLP* tools to learn complex linguistic structures like the *predicate-argument* structures previously described. The work carried out by (Briscoe and Carroll, 1997) on the automatic extraction of subcategorizing structures and some other works performed on the classification of verbs depending on the subcategorization have shown that the classification algorithms are the right computational methods to learn these types of structures.

Semantic Role Labeling (*SRL*) is nowadays one of the hottest topics in computational linguistics, and normally, the work on this task has included several probabilistic and machine-learning approaches. These kind of approaches to role labeling usually divide the task in two subtasks that consist of identifying the arguments of a predicate and the classification into their specific roles (Zapirain, 2011). In order to achieve these subtasks an architecture, shown in figure 5, consisting of three separate steps (sometimes also called *step by step SRL*) is usually followed by most state-of-the-art semantic role labeling systems.

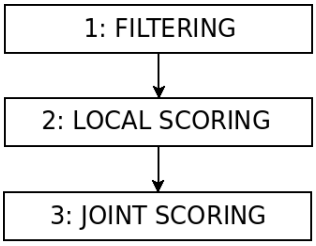

Figure 5: Steps in SRL

The first step of the process consists of **filtering** the set of argument candidates for a given predicate. The second step, on the other hand, consists of a **local scoring** of argument candidates by means of a function that outputs probabilities (or confidence scores) for each of the possible role labels (Màrquez et al., 2008). Finally, the last step consists of a **joint scoring** that produces the overall labeled argument structure for the predicate by combining the predictions from the previous step.

In addition to the boost that the development of new machine learning techniques have given to computational linguistics, and more precisely to *SRL* (Competitions like *Senseval-3* (Litkowski, 2004), *CoNLL-2005* (Carreras and Màrquez, 2005) and *SemEval-2007* (Pradhan et al., 2007) that challenge the participants to develop state-of-the-art *SRL* systems have facilitated the creation of a research community that publishes works with a high impact, such as (Gildea and Jurafsky, 2002), (Surdeanu et al., 2003)). Another key factor in the development of systems for the automatic labeling of semantic roles is the creation, in the past few years, of corpora labeled with semantic roles, like the previously described lexical databases of English *FrameNet* (Baker et al., 1998) and *PropBank* (Palmer et al., 2005) from which the *ML* algorithms in *SRL* systems can learn.

2.1 The *EPEC-ROLSEM* corpus

The Basque corpus that the labeling systems developed in this work use for learning purposes is *EPEC-ROLSEM*. This corpus is partially tagged on a semantic level according to the *VerbNet/PropBank* model. The reasons why this model was chosen over the *FrameNet* model when tagging the corpus semantically are discussed in (Aldezabal et al., 2010). *EPEC-ROLSEM* is intended to be a training corpus for the development and improvement of several *NLP* tools, as noted in (Bengoetxea and Gojenola, 2007).

The study performed on verbs taken from the *EPEC-ROLSEM* corpus in (Aldezabal, 2004) was the starting point for the semiautomatic annotation that led to the obtention of the gold standard version of the corpus used in this project. It consists of 300,000 words and 10,469 files, each corresponding to a sentence taken from standard texts written in Basque. 280 different verbs were identified in the corpus, and the number of occurrences that the verbs had were counted. The counts for each verb were analyzed and a decision was made based on this: to focus on the semiautomatic annotation of the verbs that had 30 or more occurrences in the corpus. There are 136 verbs that fulfill this condition, and the most frequent are ***Izan***, ***Egon***, and ***Hasi***. The verb senses from *VerbNet/PropBank* that correspond to the mentioned three verbs are the following: *be_01*, *be_02*, and *have_03* for *Izan*; *be_01*, *be_02*, and *correspond_02* for *Egon*; and finally *begin_01/start_01* for *Hasi*.

The files in the corpus are structured in the way shown in table 2. As it has been previously mentioned, each file contains just one sentence. These sentences are labeled at a predicate-level and contain information about the verb sense, the valence, the semantic roles and the selectional restrictions inter alia. Each argument and adjunct for the predicate is marked with an ***arg_info* tag**; the rest of the tags, on the other hand, which are marked (*ncmod*, *entios* etc.), contain information about the syntactic dependencies in the sentence. The tags that have been used in this work are the *arg_info* tags. Table 2 shows the file corresponding to the sentence *Patxi Zubizarreta idazleak irabazi zuen Antonio Maria Labaien ipuin leihaketa* (*The writer Patxi Zubizarreta won the Antonio Maria Labaien story contest*).

ncmod (-, idazleak-[w841], Zubizarreta-[w840], Zubizarreta-[w840])
entios (-, Zubizarreta-[w840], Patxi-[w839])
...
arg_info(win_01, irabazi-[w842], idazleak-[w841], arg0, Agent, -)
#w842:irabazi:IZE:ARR#w841:idazle:IZE:ARR
...
ncmod (-, lehiaketa-[w848], ipuin-[w847], ipuin-[w847])
arg_info(win_01, irabazi-[w842],lehiaketa-[w848], arg1, Theme, -)
#w842:irabazi:IZE:ARR#w848:lehiaketa:IZE:ARR
...

Table 2: File example from *EPEC-ROLSEM*

Semantic information on *EPEC-ROLSEM*

(Aldezabal et al., 2010) states that the relations [1] that are candidates to be arguments or adjuncts of the verbs are taken from the set of dependency relations associated to each clause. The *arg_info* semantic tags contain 11 fields defined as:

arg_info(VN, V[WN1], TE[WN2], VAL, VNrol, EADBrol, HM)WN1:VPred, TE_PoSKat, TE_SubKat,WN2:TEPred

arg_info(go_01, joan[w3], Mikel[w1], arg0, Agent, [+hum])#w3:joan:IZE:IZB#w1:Mikel

Table 3: *arg_info* semantic tag example from *EPEC-ROLSEM*

The mentioned fields are explained next for the example in table 3 (tag corresponding to the *ncmod* dependency between the verb *Joan (go_01)* and the argument *Mikel* in the sentence *Mikel etxera joan zen*(*Mikel went home*)).

- ***VN (VerbNet/PropBank verb)***: (*go_01*). The English verb and its *PropBank* number in *VerbNet/PropBank*.
- ***V[WN1] (Verb)***: (*joan[w3]*). Main verb, head of the relation, and the number of the word in the sentence.
- ***argument[WN2] (TE)***: (*Mikel[w1]*). The element depending on the head that will be the adjunct or the argument, and the number of the word in the sentence.
- ***VAL (Valence)***: (*arg0*). A value that identifies arguments or adjuncts (e.g. *arg0, arg1, arg2, arg3, argM*).
- ***VNrol (Role in VerbNet)***: (*Agent*). The roles usually associated with the numbered arguments and adjuncts in *PropBank (e.g. arg0: agent, experiencer,...)*.
- ***EADBrol***: The semantic role according to the *EADB* role set (e.g. theme, state, location, experiencer, ...).
- ***HM (Selectional Restriction)***: (*[+hum]*). The considered ones are *[+animate], [-animate] ([+biz], [-biz] in Basque/*EPEC-ROLSEM*), [+count], [-count] ([+kont], [-kont] in Basque/*EPEC-ROLSEM*) and [+hum], [-hum] ([+giz], [-giz] in Basque/*EPEC-ROLSEM*)*.

[1] "The relations considered are: *ncsubj*, *ncobj*, *nczobj*, *ncmod*, *ncpred* (non-clausal subject, object, indirect object, ...), *ccomp_obj*, *ccomp_subj*, *cmod* (clausal finite object, subject, modifier), *xcomp_obj*, *xcomp_subj*, *xcomp_zobj*, *xmod*, *xpred* (clausal non-finite object, subject, indirect object,...)."

- ***WN1:VPred***: (*w3:joan*). The number of the main verb in the sentence and the lemma for the main verb.
- ***TE_ PoSKat (TE's Part-of-Speech category)***: (*IZE*). Part-of-speech category of the argument.
- ***TE_ SubKat (TE's Part-of-Speech subcategory)***: (*IZB*). Part-of-Speech subcategory of the argument.
- ***WN2:TEPred***: (*w1:Mikel*). The number of the argument in the sentence and the lemma for the argument.

In the *arg_ info* tags, it is possible to have a null mark ("-") in some of the fields listed above, meaning that the annotator was not sure of the value or thought that it was not necessary to define it.

3 Experimentation and results

The methodology shown in figure 6 is followed in this section in order to cover the entire development process for the final tagger. As can be appreciated in the figure, two different approaches are initially considered. The first approach is based on linguistic rules and consists of labeling the predicate arguments by following what is established by the rules corresponding to each predicate (Aldezabal et al., 2013) and that have been previously handed to the system. The second approach is based on machine learning, and consists of building a model from a training set that will label the predicate arguments.

In order to see what the approach that will give the best results is, systems following the two approaches are developed for the three most common verbs in the *EPEC-ROLSEM* corpus, and the results are compared. The verbs are *Izan*, *Egon* and *Hasi*. This means that in total six *predicate-specific* systems have been developed for experimentation before creating the final *SRL* system that will be capable of labeling every predicate's arguments (*predicate-independent*). The final system will use the approach that gives the best results for the mentioned verbs.

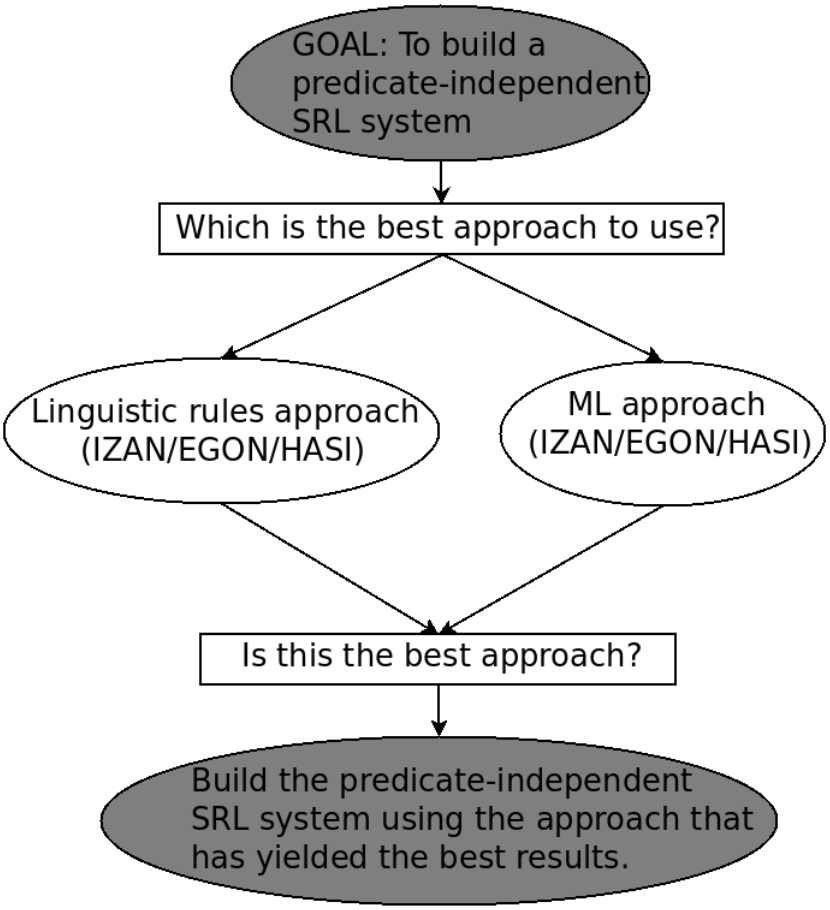

Figure 6: Project structure

The section consists of three parts, where the first one covers the linguistic rules approach, the second one covers the machine learning approach and the last part covers the development process for the final *SRL* system.

3.1 First approach: *SRL* using linguistic rules

3.1.1 Experiment description

A first approach consisting of a system that will label semantic roles based on a set of linguistic rules is considered. The verb senses for the verbs *Izan*, *Egon* and *Hasi* are presented next, as well as the rules in (Aldezabal et al., 2013) that correspond to these verbs.

Izan

The number of verb senses for the verb *Izan* are 7 as shown in figure 7. The instances that correspond to the verb *Izan* in the corpus (6796) cover %19.2 of the manually tagged instances (35379).

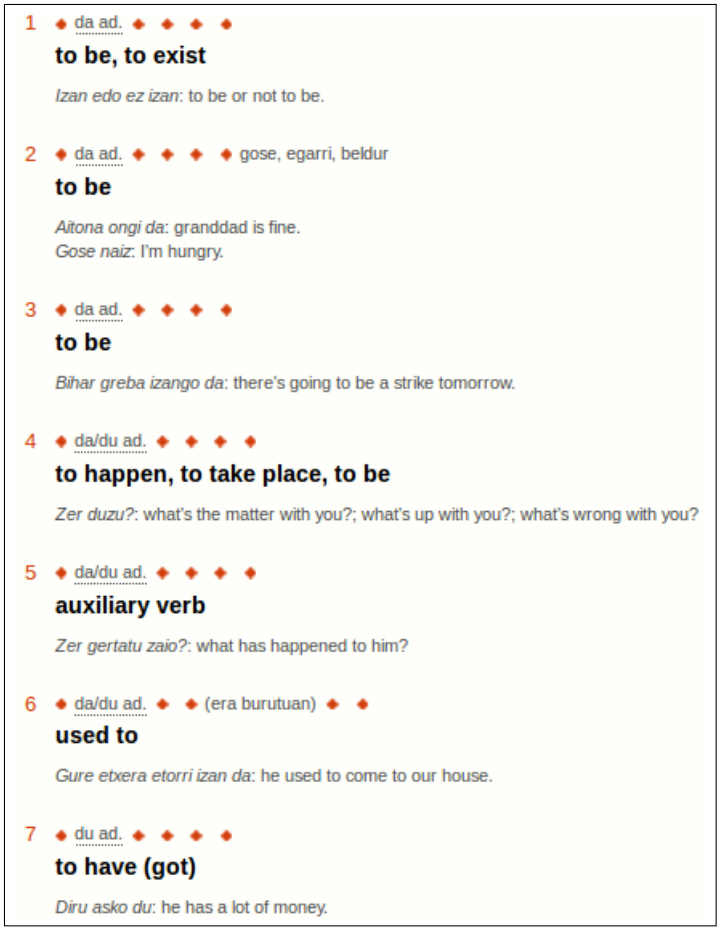

Figure 7: Verb senses for the verb *Izan*

The linguistic rules from (Aldezabal et al., 2013) map these verb senses with the verb-classes for the verb *Izan* in *VerbNet* (*be_ 01*, *be_ 02* and *have_ 03*). The rules establish the semantic role and the valence an argument should have based on the case and the roleset to which the sentence containing the argument belongs to. The rules are presented in table 4.

be_01
-*arg1: topic, gaia*-ABS/KONPL
-*arg2: attribute, ezaugarria*-ABS/KONPL
be_02
-*arg1: theme, gaia*-ABS
-*arg2: location, kokapena*-INE
have_03
-*arg0: theme, edukitzailea*-ERG
-*arg1: theme, edukia*-ABS

Table 4: Linguistic rules for *Izan*

According to these rules, if a sentence is given, for example the one in the seventh sense from figure 7: *Mikelek diru asko du.* (*Mikel has a lot of money*). The roleset from *PropBank* that corresponds to this sense of the verb *Izan* is *have_ 03* and the argument with the ergative case (*Mikelek*) should have *theme/edukitzailea* as a semantic role and *arg0* as a valence. The argument in the sentence with the absolutive case, on the other hand, should have *theme/edukia* as a semantic role and *arg1* as a valence.

It can be noticed by examining the rules that there is much room for **ambiguity** left in most of the cases. For example, if a sentence that has the predicate *Izan* and its sense is mapped in the *be_ 01* roleset, then it is not possible to precisely determine which semantic role and valence should the argument with the absolutive case be labeled with. In order to be able to implement *SRL* systems based solely on linguistic information, an effort to **disambiguate** the rules has been made by taking into account the dependency relations corresponding to the *arg_ info* semantic tags from the files in the corpus.

-**INE**→*arg2: location, kokapena*
-**ABS+ncpred**→*arg2: attribute, ezaugarria*
-**ABS+ncobj**→*arg2: attribute, ezaugarria*
-**ABS+ncsubj**→*arg1: topic, gaia*
-**KONPL**→*arg1: topic, gaia*
-**ERG**→*arg0: theme, edukitzailea*

Table 5: Inferred (disambiguated) rules for the verb *Izan*

Discriminating decisions had to be made for most of the rules described above in the disambiguation process. In fact, the systems must have the ability to make decisions and label an argument with one or another semantic role, and this has led to infer a new set of linguistic rules that have a very restricted scope and go for one or another semantic role of the ones described in the rules above. The new set of linguistic rules for *Izan* is shown in table 5.

As can be noticed, the inferred rules (new set of rules) for the verb *Izan* make a distinction between the initial rules marked with the case ABS (absolutive), based on the dependency relation. This way, none of the initial rules defined in (Aldezabal et al., 2013) for the verb *Izan*, the case being ABS, is left apart and the possibility to label this arguments correctly is significantly higher.

Egon

The number of verb senses for the verb *Egon* are 11 as shown in figure 8. The instances that correspond to the verb *Egon* in the corpus (1212) cover %3.4 of the manually tagged instances (35379).

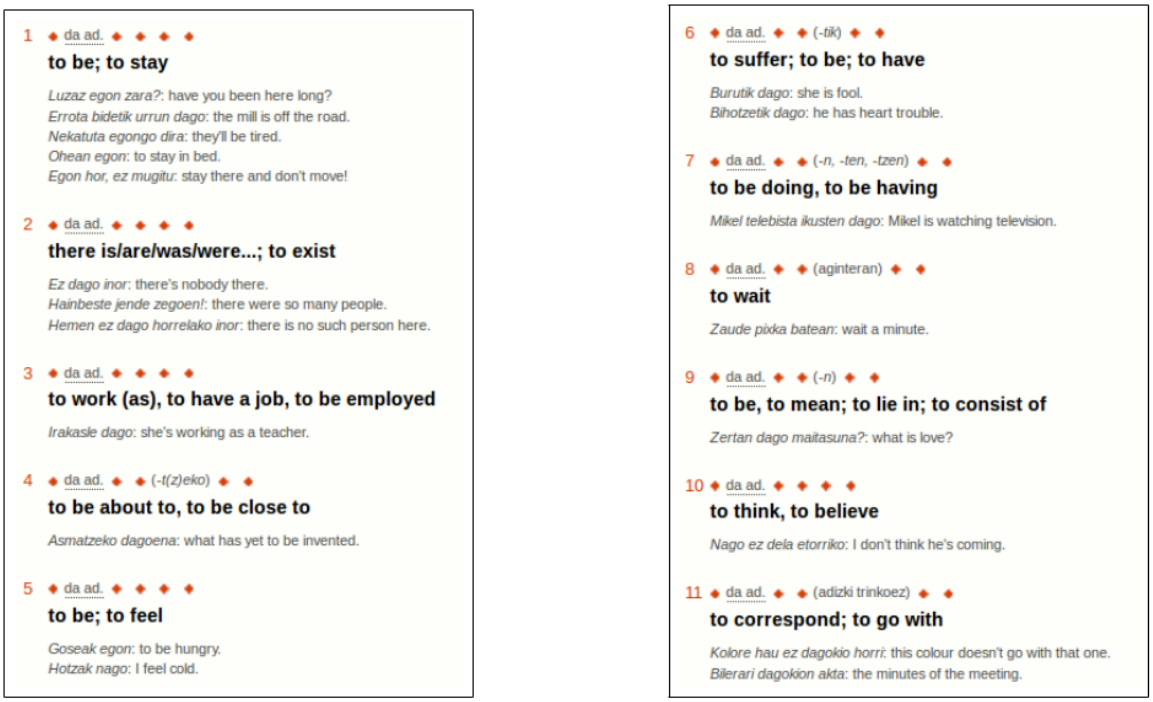

Figure 8: Verb senses for the verb *Egon*

The linguistic rules from (Aldezabal et al., 2013) map these verb senses with the **rolesets** for the verb *Egon* in *PropBank* (*be_01*, *be_02* and *correspond_02*). The rules are presented in table 6.

be_01
-*arg0: topic, gaia*-ABS/ERG
-*arg1: attribute,*
egoera-ABS/ABL/ALA/DAT/EMEN/ESPL/INS/MOD
be_02
-*arg0: theme, gaia*-ABS/PAR
-*arg1: location, kokapena*-INE/ALA/ABL
correspond_02
-*arg0: theme, gaia*-ABS
-*arg1: location, kokapena*-DAT

Table 6: Linguistic rules for *Egon*

The disambiguation process followed in *Izan* based on the the dependency relations for rules with the same case has not been followed for the verb *Egon*. The reason for this is that the dependency relations on the ambiguous cases of *Egon* are not a distinguishing factor. The case being ABS a single disambiguation has been made based on the selectional restriction [-biz] ([-animate]). The new set of linguistic rules for *Egon* is shown in table 7.

-ERG→*arg0, topic, gaia*
-ABS→*arg1, attribute, egoera*
-PAR→*arg0, theme, gaia*
-INE→*arg1, location, kokapena*
-ALA/ABL/EMEN/ESPL/INS/MOD→*arg1, attribute, egoera*
-DAT→*arg1, location, kokapena*

Table 7: Inferred (disambiguated) rules for the verb *Egon*

Hasi

The number of verb senses for the verb *Hasi* are 2 as shown in the figure 9. The instances that correspond to the verb *Hasi* in the corpus (376) cover %1 of the manually tagged instances (35379).

Figure 9: Verb senses for the verb *Hasi*

The linguistic rules from (Aldezabal et al., 2013) map these verb senses with the roleset for the verb *Hasi* in *PropBank* (*begin_ 01/start_ 01*). The rules are presented in table 8.

begin_01/start_01
-*arg0: agent, kausa*-ERG[+giz]
-*arg0: agent, esperimentatzailea*-ABS
-*arg1: theme, gai_ ukitua*-ABS[-biz]
-*arg1: theme, jarduera*-INE/SOZ
-*arg2: instrument*-INS/SOZ

Table 8: Linguistic rules for *Hasi*

The disambiguation processes made in *Izan* and *Egon* have not been made for the verb *Hasi*. The reason for this is that the dependency relations and the selectional restrictions on the ambiguous cases of *Hasi* are not distinguishing factors. The new set of linguistic rules for *Hasi* is shown in table 9.

-INS/SOZ→*arg2, instrument, -*
-INE→*arg1, theme, jarduera*
-ERG→*arg0, agent, kausa*
-ABS+[-biz]→*arg1, theme, gai_ ukitua*
-ABS→*arg0, agent, esperimentatzailea*

Table 9: Inferred (disambiguated) rules for the verb *Hasi*

It may be noted that the training set for the verb *Izan* has 6,796 instances, the set for the verb *Egon* has 1,212, and the set for the verb *Hasi* has just 376 instances.

3.1.2 System Description

Three *predicate-specific* taggers using the linguistic rules approach have been implemented (*Izan SRL*, *Egon SRL* and *Hasi SRL*). Each of them is structured in four steps and the systems use three different data-containers. The steps and the data-containers are listed below.

Steps

- ***Search occurrences***: All the files in the *Gold_ Standard* data-container are read and searched for occurrences of the verb being labeled.

- ***Erase***: All the files that are found in the *Gold_ Standard* data-container are copied into the *Erased* data-container. Then, the argument and the semantic roles from the *arg_ info* tags found for the verb being labeled in the files of the data-container *Erased* are erased.

- ***Label***: All the files that are found in the *Erased* data-container are copied into the *Labeled* data-container. Then, the argument and the semantic roles for the *arg_ info* tags found for the verb being labeled and that have been previously erased in the files of the data-container *Labeled* are labeled. The labeling is done using the inferred heuristics that are shown in the previous tables.

- ***Compare***: Each file in the *Gold_ Standard* data-container is read and compared to the corresponding file in the *Labeled* data-container. The precision, the recall and the F-measure for the arguments of the verb being labeled are calculated.

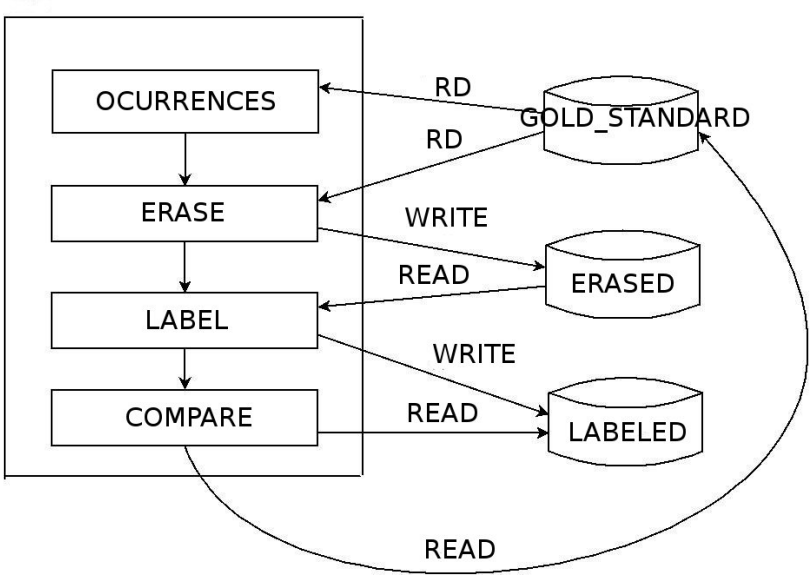

Figure 10: Architecture of the linguistic rules taggers

Data-containers

- ***Gold_ Standard***: Stores the 10,469 files corresponding to the gold standard version of the *EPEC-ROLSEM* corpus.

- ***Erased***: Stores 10,469 files where the argument and the semantic roles from the *arg_ info* tags of the verb being labeled have been erased.

- ***Labeled***: Stores 10,469 files where the argument and the semantic roles from the *arg_ info* tags of the verb being labeled have been labeled using the inferred heuristics.

3.1.3 Results

The results obtained from the three taggers using the approach treated in this subsection are shown next. These results will later be compared to the scores obtained from the three other taggers using the machine learning approach in order to decide which is the best method to use by the final semantic role tagger for Basque verbs. It may be noted that the linguistic rules used to implement the systems do not establish how adjuncts (*argM*) should be labeled. This is the reason why all the instances that do not get labeled with what is indicated in the rules gets automatically assigned the *argM* tag and causes getting a very low precision for adjuncts.

Izan				
Role	**Count**	**Precision**	**Recall**	**F-Measure**
arg0	249	0.984	0.731	0.839
arg1	2292	0.983	0.721	0.832
arg2	2532	0.740	0.737	0.738
argM	1723	0.567	0.793	0.661
OVERALL	6796	0.787	0.745	0.753

Table 10: Results of the linguistic-rule approach for *Izan*

Egon				
Role	**Count**	**Precision**	**Recall**	**F-Measure**
arg0	332	0.946	0.684	0.794
arg1	710	0.881	0.458	0.602
argM	170	0.201	0.729	0.315
OVERALL	1212	0.805	0.557	0.615

Table 11: Results of the linguistic-rule approach for *Egon*

Hasi				
Role	**Count**	**Precision**	**Recall**	**F-Measure**
arg0	45	0.6	0.733	0.660
arg1	168	0.684	0.774	0.726
arg2	10	0.444	0.8	0.571
argM	153	0.830	0.612	0.705
OVERALL	376	0.726	0.704	0.705

Table 12: Results of the linguistic-rule approach for *Hasi*

Table 10 shows that the argument that has the biggest number of occurrences is *arg2* (2532), closely followed by *arg1* (2292). Regarding the precision, it may be noted that in

general the values obtained are very high (0.984, 0.983 and 0.740). Table 11; on the other hand, shows that the best f-Measure value obtained is the one for argument *arg0* (0.794). The f-measure values for this tagger are lower than the ones obtained for the previous one that corresponded to the verb *Izan*. Finally, the results in table 12 show that the tagger for the verb *Hasi* is the one that labels the smallest number of occurrences (376) compared to the previous taggers that label 6796 and 1212 occurrences respectively.

3.2 Second approach: SRL using machine learning

3.2.1 Experiment description

As is stated in previously, three *SRL* systems corresponding to the Basque verbs *Izan*, *Egon* and *Hasi* have been developed using the machine learning approach. This subsection covers the process followed to develop the three systems.

In order to be able to implement semantic role labeling systems using machine learning techniques it is necessary to first identify features that will provide significant information and will help the learning algorithms choose between the right class. As stated in (Xue and Palmer, 2005) one characteristic of feature-based semantic role modeling is that the feature space is generally large in contrast to the low-level *NLP* tasks such as *Part-Of-Speech* tagging, which generally have a small feature space. The 12 features that have been used to build the three *predicate-specific* machine learning *SRL* systems are shown below.

Features

- ***Lemma for the argument***: Holds the lemma for the element that fulfills the argument being treated in the *arg_ info* tag. It is a string-type feature.
- ***Part-Of-Speech category for the argument***: Holds the *Part-Of-Speech* category for the element that fulfills the argument being treated in the arg_info tag. The number of different Part-Of-Speech categories that have been identified is 12. It is a nominal-type feature.
- ***Part-Of-Speech subcategory for the argument***: Holds the Part-Of-Speech subcategory for the element that fulfills the argument being treated in the *arg_ info* tag. The number of different Part-Of-Speech subcategories that have been identified is 16, plus a subcategory that has been created and named *"EMPTY"* for those arguments whose Part-Of-Speech category does not have a subcategory. It is a nominal-type feature.
- ***Case for the argument***: Holds the case for the element that fulfills the argument being treated in the *arg_ info* tag. The number of different cases that have been identified is 17, plus a null mark (*"-"*) case that has been created for those arguments whose case is not defined. In addition, a case named *CONBCASE* has been

created for those arguments whose case belongs to a special set of cases identified in (Aldezabal et al., 2013). The number of different cases in the special set is 54. It is a nominal-type feature.

- ***Syntactic function for the argument***: Holds the syntactic function for the element that fulfills the argument being treated in the *arg_ info* tag. The number of different syntactic functions that have been identified is 3, *subj*, *obj* and *zobj* plus a null mark (*"-"*) syntactic function that has been created for those arguments whose syntactic function is not defined. It is a nominal-type feature.

- ***Position of the argument according to the position of the predicate***: Holds the position of the element that fulfills the argument being treated in the *arg_ info* tag according to the position of the predicate. The number of different positions that have been identified is 2, *before* and *after*. It is a nominal-type feature.

- ***Distance in number of words between the argument and the predicate***: Holds the absolute distance in number of words between the element that fulfills the argument being treated in the *arg_ info* tag and the predicate. It is a numeric-type feature.

- ***Distance in number of arguments between the argument and the predicate***: Holds the distance in number of arguments between the element that fulfills the argument being treated in the *arg_ info* tag and the predicate. It is a numeric-type feature.

- ***Frame***: Holds the *predicate-argument* structure for the proposition to which the *arg_ info* tag belongs (e.g. *arg_ PRED_ arg_ arg*). It is a string-type feature.

- ***Dynamic-Frame***: Holds the *predicate-argument* structure for the proposition to which the *arg_ info* tag belongs and marks the argument being treated by upper-casing it. It is a string-type feature (e.g. *arg_ PRED_ arg_ ARG* if the *arg_ info* tag being treated is the one corresponding to the third argument of the *predicate-argument* structure).

- ***Name entity***: Holds the entity to which the element that fulfills the argument being treated in the *arg_ info* tag belongs. The number of different entities that have been identified is 3, *Place*, *Organization* and *Person*. The null mark *"-"* will be attached to elements with no entity information marked. It is a nominal-type feature.

- ***Number entity***: Holds the kind of number entity the argument being treated in the *arg_ info* tag is in case the argument is a numeric value, e.g. date, price. It is a nominal-type feature.

The class to be predicted by the classifier using the mentioned features is shown next:

- ***Class***: Different classes that can be predicted are: *arg0, arg1, arg2, arg3, ..., argM*TMP, argM*LOC, ...* etc.

Feature constraints

In order to have features that will really help the learning algorithms predict one or another class, constraints have been established for the features that are most likely to have a wide range of different possible values. The features with a high distribution factor that have been constrained are the *lemma for the argument* and the *Distance in number of words between the argument and the predicate*.

The constraint set for the lemma establishes that if the lemma identified for the *arg_ info* tag being treated has an overall occurrence (taking into account all the tags corresponding to the verb in the gold standard version of *EPEC-ROLSEM*) greater than 2 it will be considered. Otherwise, it will not be taken into account. The constraint settled for the distance in number of words, on the other hand, establishes that if the distance is greater than 12 words a special tag that indicates this condition will be set for the distance. Otherwise, the real distance will be considered.

Machine learning

The *ML* systems have been tested using algorithms of different types. Using several classifiers for testing makes it possible to know which the learning-paradigm that best suits the semantic role labeling task is.

The technique used to estimate how accurate the models developed will be in a real-world environment is *cross-validation*. This technique has been chosen over the *train-test* technique due to the contained size of the training sets available. The training set for the verb *Izan* has 6,796 instances, the set for the verb *Egon* has 1,212, and the set for the verb *Hasi* has just 376 instances.

Regarding the number of folds used to perform cross-validation, 3, 5, 10, 12 and 15 folds were used for experimentation purposes. The results for the **10-fold cross-validation** have been taken as reference values. According to (Witten et al., 1999) extensive tests on numerous datasets, with different learning techniques, have shown that 10 is about the right number of folds to get the best estimate of error, and there is also some theoretical evidence that backs this up.

Classifier types

Five learning algorithms have been used in this work, two tree-type classifiers (*J48* and *Random Forest*), a classifier based on functions (*SMO*), a Bayesian one (*NaïveBayes*) and, finally, a lazy-type algorithm (*IBK*). The *Random Forest* and *SMO* classifiers are well known to have a good performance regarding *NLP* tasks in general.

- ***SMO (*Sequential Minimal Optimization*)***: *SVM* (*Support Vector Machine*) implementation created by John Platt (Platt et al., 1998). *SMO* breaks the problem into a series of smallest possible sub-problems, which are then solved analytically.

- ***IBK***: Commonly known as the *K-NN* (*K-Nearest Neighbor*) classifier. Gives the possibility to set different values for K and to establish a weight function that depends on distance.

- ***Random Forest***: Classifier developed by Leo Breiman (Breiman, 2001). Operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees.

- ***J48***: *Java* implementation of the *C4.5* algorithm (Quinlan, 1993). *C4.5* was developed by Ross Quinlan and is used to create decision trees. *C4.5* is an extension of the *ID3* algorithm previously developed by Quinlan.

- ***NaïveBayes***: Probabilistic classifier that applies the *Bayes* formula.

3.2.2 System Description

The machine learning systems implemented divide the process into two different stages as shown in figure 11. In the first stage, the features for the semantic tags that correspond to the verb being labeled in the system (*predicate-specific*) are extracted, and then, in the second stage, the training set is created using the information from the previous stage and the classifier is built.

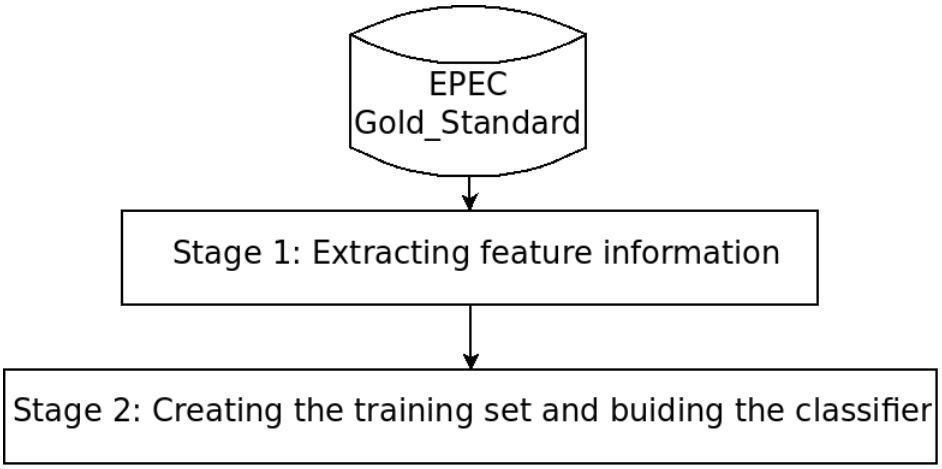

Figure 11: *Machine learning SRL* stages

arg_ info feature files

In order to feed the second stage with the feature information of each semantic tag that corresponds to the verb being labeled (*Izan*, *Egon* or *Hasi*), the application will create *arg_ info feature files* for the tags. The *arg_ info* feature file for the ***arg_ info(begin_ 01/start_ 01, hasi[w622], beldurtzen[w621], arg1, Theme)#w622: hasi: ADI: SIN #w621: beldurtu*** semantic tag is shown in figure 12.

File: File1.txt
Arg_Info Number: 2

-Lemma for the treated element: Beldurtu
-POS category: ADI
-POS subcategory: SIN
-Case: mod
-Syntactic function: SUBJ
-Position of treated elem. according to the predicate:before
-Distance words: 1
-Distance arguments: 1
-Frame: arg_arg_PRED_arg_arg
-Dynamic-Frame: arg_ARG_PRED_arg_arg
-Name entity: -
-Number entity: -

-Class: arg1

Figure 12: *arg_ info* feature file

Three taggers using the *ML* approach have been implemented (*Izan SRL*, *Egon SRL* and *Hasi SRL*). The application is structured in four parts as shown in figure 13. In addition, the system uses two different data-containers and some additional steps. All of them are listed below.

Parts

- ***Methods***: Holds a set of methods that are used by the other parts of the application (Stages 1 and 2).
- ***CreateDataSet***: Creates the training file to be used by *Learning* (Stage 2).
- ***Learning***: Reads the training file and filters it, builds the classifier and gets the results (Stage 2).
- ***Main***: Runs the system (Stages 1 and 2).

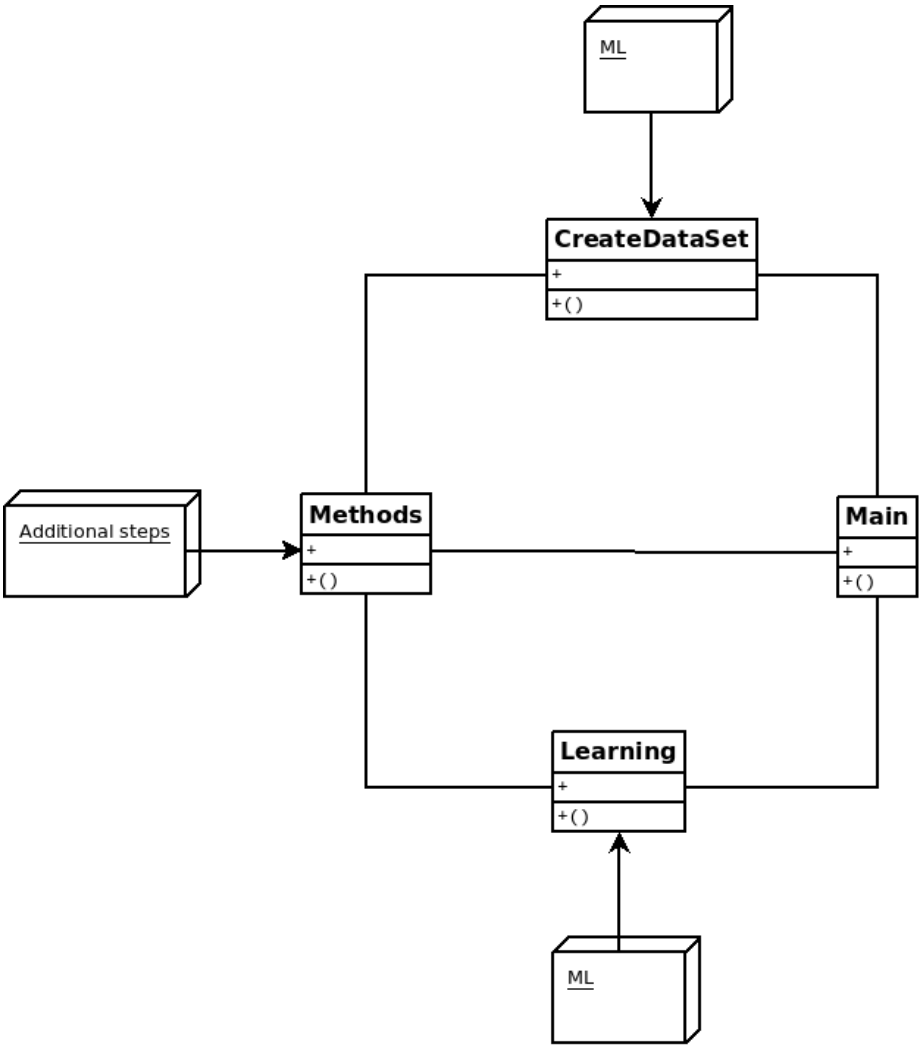

Figure 13: Architecture of the machine learning approach tagger

Steps

- ***Search occurrences***: All the files in the *Gold_Standard* data-container are read and searched for occurrences of the verb being labeled (Stage 1).
- ***Create feature files***: *arg_info feature files* for each occurrence of the verb being labeled are created and stored in the *Features* data-container (Stage 1).
- ***Count occurrences***: The number of occurrences that different values adopted by each of the features have are counted. This will later be used to apply constraints (Stage 1).
- ***Apply constraints***: Constraints in the feature files for the lemma and the distance in words are applied (Stage 1).

Data-containers

- ***Gold_ Standard***: Stores the 10,469 files corresponding to the gold standard version of the *EPEC-ROLSEM* corpus.
- ***Features***: Stores the *arg_ info feature files* corresponding to the *arg_ info* tags of the verb being labeled.

3.2.3 Results

The results obtained for all the learning algorithms that have been tested over different numbers of folds are shown in tables 13, 14, 15.

Izan					
F-Measure	**SMO**	**IBK**	**R.For.**	**J48**	**N.B.**
3-fold	0.851	0.8	0.83	0.846	0.797
5-fold	0.851	0.802	0.826	0.846	0.798
10-fold	0.851	0.801	0.827	0.854	0.8
12-fold	0.85	0.8	0.828	0.849	0.798
15-fold	0.851	0.796	0.827	0.851	0.801

Table 13: Classifiers for *Izan*

Egon					
F-Measure	**SMO**	**IBK**	**R.For.**	**J48**	**N.B.**
3-fold	0.83	0.818	0.821	0.843	0.784
5-fold	0.824	0.827	0.827	0.837	0.796
10-fold	0.829	0.825	0.831	0.846	0.805
12-fold	0.83	0.823	0.824	0.839	0.808
15-fold	0.833	0.23	0.833	0.841	0.804

Table 14: Classifiers for *Egon*

Hasi					
F-Measure	**SMO**	**IBK**	**R.For.**	**J48**	**N.B.**
3-fold	0.667	0.652	0.66	0.686	0.594
5-fold	0.67	0.652	0.684	0.666	0.617
10-fold	0.663	0.623	0.647	0.673	0.624
12-fold	0.657	0.664	0.679	0.671	0.608
15-fold	0.671	0.655	0.657	0.661	0.623

Table 15: Classifiers for *Hasi*

It can be noticed by the tables showing the results for the different algorithms that the classifier that best suits the semantic role labeling task out of the ones that have been used for experimentation is *J48*, followed by the function type *SMO* algorithm. On the other end, the one with the worst performance has proved to be *NaïveBayes*. Regarding the number of folds to be used when performing cross-validation, the best results have been generally achieved by using 10 and 12-folds.

The results obtained from the three taggers using the approach treated in this subsection are shown in tables 16, 17 and 18. These tables show the precision, the recall and the f-measure values for each argument and the overall f-measure value shown is the best from the values obtained in tables 13, 14 and 15. The results from the below tables are comparable to the scores obtained from the three other taggers using the linguistic rules approach that are shown in tables 10, 11 and 12. The values on the tables correspond to a 10-fold cross-validation using the *J48* classifier.

Izan (CV-10, J48)				
Role	**Count**	**Precision**	**Recall**	**F-Measure**
arg0	249	0.895	0.855	0.875
arg1	2292	0.949	0.939	0.944
arg2	2532	0.875	0.9	0.888
argM*LOC	211	0.385	0.427	0.404
argM*TMP	406	0.524	0.679	0.592
argM*MNR	273	0.749	0.601	0.667
argM*CAU	103	0.877	0.903	0.89
argM*ADV	66	0.091	0.015	0.026
argM*PRP	89	0.932	0.921	0.927
argM*-	262	0.859	0.753	0.803
argM*NEG	255	0.992	1	0.996
argM*DIS	58	0.688	0.393	0.5
OVERALL	6796	0.854	0.857	0.854

Table 16: Results of the machine learning approach for *Izan (CV-10, J48)*

The results in table 16 show that the argument (core-argument) that gets labeled the best (i.e. the one that has the highest f-measure value) from the ones considered in the *predicate-specific ML* system for the verb *Izan* is *arg1* (0.944) followed by *arg2* (0.888) and *arg0* (0.875).

Egon (CV-10, J48)				
Role	**Count**	**Precision**	**Recall**	**F-Measure**
arg0	332	0.852	0.919	0.884
arg1	710	0.888	0.907	0.898
argM*TMP	38	0.611	0.579	0.595
argM*MNR	23	0.353	0.261	0.3
argM*CAU	11	0.9	0.818	0.857
argM*PRP	8	1	0.875	0.933
argM*-	61	0.412	0.269	0.326
argM*NEG	29	1	1	1
OVERALL	1212	0.838	0.856	0.846

Table 17: Results of the machine learning approach for *Egon (CV-10, J48)*

The results in table 17, on the other hand, show that the argument that gets labeled the best from the ones considered in the *predicate-specific ML* system for the verb *Egon* is *arg1* (0.898) followed by *arg0* (0.884).

Hasi (CV-10, J48)				
Role	**Count**	**Precision**	**Recall**	**F-Measure**
arg0	45	0.868	0.733	0.795
arg1	168	0.751	0.792	0.771
arg2	10	0.4	0.4	0.4
argM*LOC	20	0.538	0.35	0.424
argM*TMP	69	0.495	0.739	0.593
argM*MNR	30	0.6	0.3	0.4
argM*CAU	8	0.857	0.75	0.8
argM*PRP	7	0.667	0.571	0.615
argM*-	14	1	0.333	0.5
argM*NEG	5	1	1	1
OVERALL	376	0.698	0.682	0.673

Table 18: Results of the machine learning approach for *Hasi (CV-10, J48)*

Finally, the results in table 18 show that the core-argument that gets labeled with the highest f-measure value from the ones considered in the *predicate-specific ML* system for the verb *Hasi* is *arg0* (0.795) followed by *arg1* (0.771) and *arg2* (0.4). In all three systems the adjunct that gets labeled with the highest precision is *argM*NEG* by far (0.992 in *Izan* and 1 in *Egon* and *Hasi*).

The next table (19) summarizes (compares) the f-measure values obtained for each verb (*Izan*, *Egon* and *Hasi*) and each argument using the linguistic rules and the machine learning approach.

F-Measure for both approaches						
	Izan		**Egon**		**Hasi**	
Role	**LR**	**ML**	**LR**	**ML**	**LR**	**ML**
arg0	0.839	0.875	0.794	0.884	0.66	0.795
arg1	0.832	0.944	0.602	0.898	0.726	0.771
arg2	0.738	0.888	-	-	0.571	0.4
argM*LOC	-	0.404	-	-	-	0.424
argM*TMP	-	0.592	-	0.595	-	0.593
argM*MNR	-	0.667	-	0.3	-	0.4
argM*CAU	-	0.89	-	0.857	-	0.8
argM*ADV	-	0.026	-	-	-	-
argM*PRP	-	0.927	-	0.933	-	0.615
argM*-/argM	0.661	0.803	0.315	0.326	0.705	0.5
argM*NEG	-	0.996	-	1	-	1
argM*DIS	-	0.5	-	-	-	-
OVERALL	0.753	0.854	0.615	0.846	0.705	0.673

Table 19: F-Measure values for both approaches

As can be noticed in table 19 the results obtained for adjuncts are classified in different types only in the systems using machine learning. The reason for this is that the linguistic rules used in the other approach do not establish how adjuncts should be labeled. In this approach, all the arguments that can not be labeled with what is established in the rules is automatically labeled with the *argM* adjunct tag. This is the reason for the bad results adjuncts have in the linguistic rules approach to *SRL*. In addition, it may be noted that the values obtained by using the machine learning approach are better for the three verbs when labeling *arg0* and *arg1* arguments.

3.3 Final SRL System

This subsection covers the development process for the final semantic role tagger. As it has been previously noted this tagger will label arguments independently of the predicate (*predicate-independent*).

3.3.1 Choosing the best approach

As previously noted in subsection 3, the final *SRL* system that will be capable of labeling every predicate's arguments (*predicate-independent*) will be implemented using the approach that gives the best results for the verbs *Izan*, *Egon* and *Hasi* (see table 19).

Regarding the linguistic rules approach, it can be noticed that the heuristics used to implement the system do not establish how exactly *adjunct-type* arguments should be labeled, and this directly affects the overall results and the number of *arg_ info* tags that

can be automatically annotated by using only linguistic rules. In fact, for the 6,796 tags manually tagged that the verb *Izan* has in the gold standard *EPEC-ROLSEM*, only 4,385 corresponding to arguments that are not adjunct-like are labeled automatically. For the verb *Egon* 680 are automatically labeled out of 1212 manually tagged tags, and for the verb *Hasi* just 179 tags out of 375 are labeled. The *f-measure* values obtained go from around 0.6 (*Hasi* and *Egon*) to a maximum of approximately 0.8 (*Izan*).

It can also be noticed by comparing the results obtained for the verb *Izan* compared to the results of the other two verbs, that disambiguating the linguistic rules completely is a key factor in order to obtain good labeling results. The discriminating decisions made in order for the system to be disambiguated for the verbs *Egon* and *Hasi* have resulted in a success rate considerably lower than the one for *Izan*.

Regarding the machine learning approach, it can be noticed that all three verbs have similar results that go from an *f-measure* value of almost 0.8 to an approximate value of 0.9 for the arguments that are not adjunct-like. In addition, the systems using the *ML* approach make a distinction between different adjunct-type elements, as opposed to the ones using the linguistic rules approach.

Therefore, due to the scores for the machine learning approach being in all cases 0.1-0.2 points higher than the scores obtained from the linguistic rules approach, and due to the ability to label adjunct-like arguments according to the type (of adjunct) when using *ML*, it has been decided that the final *SRL* tagger will use the *ML* approach.

3.3.2 System Description

As previously stated in 3.2.1, in order to be able to implement a semantic role labeling system using *ML* techniques, it is necessary to first identify the features that will guide the classifier in the selection process for the right class. The systems previously developed for *Izan*, *Egon* and *Hasi* dealt with 12 features; now, an additional feature has been added: the lemma of the predicate.

If the previous systems had been *predicate-specific*, this feature would have had the same value in all the instances from the training sets used. Nevertheless, the predicate's lemma becomes a very significant feature when building a *predicate-independent SRL* application and will provide the learning algorithm with a great amount of useful information (see figures 14 and 15).

Bonus feature

- ***Lemma for the predicate***: Holds the lemma for the predicate of the proposition corresponding to the element being treated in the *arg_ info* tag. It is a string-type feature.

Bonus feature constraint

As previously mentioned, constraints have been set to some features with a wide range of values in order to debug and cluster (as far as possible) the data. The recently added predicate lemma is a string-type feature; this means that it can adopt many distinct values. As stated in subsection 2.1, the number of different predicates present in the *EPEC-ROLSEM* corpus is 280 and the number of different predicates that have been tagged is 136 (the ones that have 30 or more occurrences in the corpus). This means that %64.5 of the corpus (35379 instances) has been manually tagged. 136 being a relatively big number, a constraint has been set for debugging purposes and will be applied to the *arg_ info* features created for the *predicate-independent* system being developed.

arg_ info feature files

Since the final system uses the same approach as the systems described in subsection 3.2, the architecture followed by this system will be the (same) one shown in figure 13. It has been noted in 3.2.2 that in order to feed the second stage of the system shown in 11 with the feature information of each semantic tag, the application will create *arg_ info feature* files. The *arg_ info* files for the final tagger will look like the ones shown in the following boxes.

File: File1.txt
Arg_Info Number: 2

-Lemma for the predicate: Joan
-**Lemma for the treated element:** Ainf_Lema
-**POS category:** ADI
-**POS subcategory:** SIN
-**Case:** mod
-**Syntactic function:** -
-**Position of treated elem. according to the predicate:**before
-**Distance words:** 1
-**Distance arguments:** 1
-**Frame:** arg_arg_PRED_arg_arg
-**Dynamic-Frame:** arg_ARG_PRED_arg_arg
-**Name entity:** -
-**Number entity:** -

-**Class:** arg1

Figure 14: new *arg_ info* feature file (1)

File: File3.txt
Arg_Info Number: 1

-Lemma for the predicate: Hartu
-**Lemma for the treated element:** Ainf_Lema
-**POS category:** ADI
-**POS subcategory:** SIN
-**Case:** mod
-**Syntactic function:** -
-**Position of treated elem. according to the predicate:**before
-**Distance words:** 1
-**Distance arguments:** 5
-**Frame:** arg_PRED_arg_arg
-**Dynamic-Frame:** ARG_PRED_arg_arg
-**Name entity:** -
-**Number entity:** -

-**Class:** arg0

Figure 15: new *arg_ info* feature file (2)

3.3.3 Results

The results obtained for the final *SRL* system using all the learning algorithms and tested over different numbers of folds are shown in table 20.

(Predicate-independent) SRL system (CV-10, J48)					
F-Measure	**SMO**	**IBK**	**R.For.**	**J48**	**N.B.**
3-fold	0.795	0.724	0.757	0.797	0.714
5-fold	0.804	0.729	0.759	0.808	0.719
10-fold	0.808	0.731	0.762	0.811	0.721
12-fold	0.809	0.73	0.764	0.812	0.721
15-fold	0.809	0.731	0.763	0.812	0.723

Table 20: Classifiers for the final system (CV-10, J48)

The results for each role of the final *SRL* system that uses the *ML* approach are shown in table 21. The values on the table correspond to a 10-fold cross-validation that has used the *J48* classifier.

(Predicate-independent) SRL system (CV-10, J48)				
Role	**Count**	**Precision**	**Recall**	**F-Measure**
arg0	4876	0.99	0.916	0.937
arg1	12594	0.923	0.934	0.929
arg2	5062	0.761	0.775	0.768
arg3	397	0.67	0.587	0.626
argM*LOC	2197	0.588	0.77	0.667
argM*TMP	3249	0.762	0.657	0.706
argM*MNR	2546	0.587	0.67	0.626
argM*CAU	514	0.778	0.833	0.805
argM*ADV	685	0.475	0.396	0.432
argM*PRP	449	0.701	0.82	0.756
argM*-	1409	0.667	0.476	0.555
argM*NEG	1078	0.983	0.993	0.988
argM*DIS	199	0.386	0.196	0.26
argM*DIR	26	0.453	0.192	0.27
argM*MOD	98	0.43	0.439	0.434
OVERALL	35379	0.819	0.814	0.812

Table 21: Results for the final tagger (CV-10, J48)

The results in table 21 show that the argument (core-argument) that gets labeled the best (highest f-measure value) from the ones considered in the *predicate-independent* system is *arg0* (0.937) followed by *arg1* (0.929), *arg2* (0.768) and *arg3* (0.626). Nevertheless, it must be taken into account that the number of arguments assigned with the *arg0* tag are less (4876) than the ones assigned with the *arg1* tag (12594).

Regarding adjuncts, it may be noticed that the adjunct with the highest precision, recall and f-measure values is *argM*NEG* (0.983, 0.993 and 0.988 respectively). The reason for this fact lies on the easily predictable syntactic position that this type of adjuncts

usually have within sentences. The adjuncts with the worst results, on the other hand, are *argM*DIR* and *argM*DIS* with 0.26 and 0.27 f-measure values. These very low values are a result of the few occurrences these adjuncts have in the corpus (*EPEC-ROLSEM*) used to train the system (199 and 26 labels). The total number of arguments and adjuncts labeled using the *SRL* system is 35379, where the most frequent is the *arg1* tag (12594 labels) and the less frequents is the *argM*DIR* tag (26 labels).

4 Conclusions and future works

As a result of the work, a *predicate-independent SRL* system have been developed. It can be concluded that the technique that best suits the task of labeling semantic roles, in this case for basque verbs, is machine learning. The *ML* approach not only offers better results but also gives the possibility to label most of the adjunct-like arguments with quite a high precision. As for the exact results obtained in the final system, it can be stated that the *f-measure* values for arguments *arg0*, *arg1*, *arg2* are acceptable and indeed quite good. Regarding adjunct-like arguments, negatives (*argM*NEG*) are the ones with the best result (0.988) and *argM*DIS* adjuncts are the ones with the worst results (0.26). The reason why some arguments have doubtful scores (e.g *argM*DIS* or *argM*DIR*) is that there are too few *arg_info* semantic tags that correspond to those arguments in the *EPEC-ROLSEM* corpus for the learning algorithm to be effective. The best overall *f-measure* value is 0.812, as can be seen in table 20, and has been obtained by performing a 12-fold (or 15-fold) cross-validation test using the *J48* tree-type algorithm. The worst overall *f-measure* values have been given by the *NaïveBayes* algorithm on a 3-fold cross-validation test (0.714).

Encountered problems

The most significant problem when developing this paper was dealing with the ambiguous linguistic rules in section 3.1. It is explained there that the need to be disambiguated and to have normally just the *case* to distinguish between different semantic roles has led to infer rules that have a very restricted scope and go for one or another semantic role. The way this problem has been handled is also shown in section 3.1.

Another issue that had to be taken care of in addition to the mentioned one is the problem with the *arg_info* tags from the *EPEC-ROLSEM* corpus that were supposed to be labeled and were not. When the gold standard corpus was provided in the early stages of the work here presented, it was specified that all the semantic tags except some corresponding to the verbs *Izan*, *Egon* and *Hasi* had been previously tagged manually. This turned out not to be true; semantic tags that correspond to other verbs were also found. This was handled by not taking into account the unlabeled *arg_info* semantic tags.

Finally, some other very specific problems that came from irregularities in the semiautomatic labeling of the corpus were encountered. These include semantic tags that contained words with no identifier, files with *arg_info* tags repeated several times, etc. These problems were located within the corpus and were manually corrected.

Future works

There are several future tasks that could be performed in order to get better results for both approaches. Considering the linguistic rules approach, the heuristics used could be improved and somehow disambiguated, based on new linguistic information other than the case or the dependency relations, and new rules could be written in order to be able to

label adjunct-like arguments. Considering the machine learning approach, new features could be identified in order to get better results for the arguments with poor scores.

The idea of developing a hybrid semantic role labeling system where linguistic rules would label arguments in very specific cases, in which the machine learning system failed or had a low success rate, could be also considered. A previous analysis should be performed to identify these particular cases.

References

Aldezabal. *Aditz-azpikategorizazioaren azterketa sintaxi partzialetik sintaxi osorako bidean. 100 aditzen azterketa, Levin-en (1993) lana oinarri hartuta eta metodo automatikoak baliatuz*. PhD thesis, PhD thesis, University of the Basque Country, 2004.

Aldezabal, Aranzabe, de Ilarraza, and Estarrona. Building the basque propbank. *Proceedings of LREC-2010*, 2010.

Aldezabal, Aranzabe Urruzola, Díaz de Ilarraza Sánchez, and Estarrona Ibarloza. A methodology for the semiautomatic annotation of epec-rolsem, a basque corpus labeled at predicative level following the propbank-verb net model. *UPV/EHU/LSI/TR; 01-2013*, 2013.

Baker, Fillmore, and Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

Bengoetxea and Gojenola. Desarrollo de un analizador sintáctico estadístico basado en dependencias para el euskera. *Procesamiento del Lenguaje Natural*, 1(39):5–12, 2007.

Boas. Bilingual framenet dictionaries for machine translation. In *LREC*, 2002.

Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Briscoe and Carroll. Automatic extraction of subcategorization from corpora. In *Proceedings of the fifth conference on Applied natural language processing*, pages 356–363. Association for Computational Linguistics, 1997.

Carreras and Màrquez. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 152–164. Association for Computational Linguistics, 2005.

Fillmore, Ruppenhofer, and Baker Collin F. Framenet and representing the link between semantic and syntactic relations. *Frontiers in linguistics*, 1:19–59, 2004.

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

Kingsbury and Palmer. Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, volume 3, 2003.

Kipper, Dang Trang, and Palmer. Class-based construction of a verb lexicon. In *AAAI/IAAI*, pages 691–696, 2000.

Schuler Karin Kipper. Verbnet: A broad-coverage, comprehensive verb lexicon. 2005.

Levin. *English verb classes and alternations: A preliminary investigation*, volume 348. University of Chicago press Chicago, 1993.

Litkowski. Senseval-3 task: Automatic labeling of semantic roles. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, volume 1, pages 141–146, 2004.

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

Melli, Wang, Yang, and Liu. Description of squash, the sfu question answering summary handler for the duc-2005 summarization task. *safety*, 1:14345754, 2005.

Palmer. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the Generative Lexicon Conference*, pages 9–15, 2009.

Palmer, Gildea, and Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

Platt et al. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

Pradhan, Loper, Dligach, and Palmer. Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87–92. Association for Computational Linguistics, 2007.

Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

Shen and Lapata. Using semantic roles to improve question answering. In *EMNLP-CoNLL*, pages 12–21, 2007.

Surdeanu, Harabagiu, Williams, and Aarseth. Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 8–15. Association for Computational Linguistics, 2003.

Swier and Stevenson. Unsupervised semantic role labelling. In *Proceedings of EMNLP*, volume 95, page 102, 2004.

Witten, Frank, Trigg, Hall, Holmes, and Cunningham. Weka: Practical machine learning tools and techniques with java implementations, 1999.

Xue and Palmer. Automatic semantic role labeling for chinese verbs. In *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 19, page 1160. Citeseer, 2005.

Zapirain. *Rol semantikoen etiketatze automatikoa: rol multzoak eta hautapen murriztapenak*. PhD thesis, Euskal Herriko Unibertsitatea, 2011.