Procesamiento del Lenguaje Natural, Revista nº 50 marzo de 2013, pp 61-68 recibido 30-11-12 revisado 07-02-13 aceptado 19-02-13

Transforming Complex Sentences using Dependency Trees for Automatic Text Simplification in Basque

Transformación de las oraciones compuestas utilizando árboles de dependencias para la Simplificación Automática de Textos en Euskera

María Jesús Aranzabe, Arantza Díaz de Ilarraza, Itziar Gonzalez-Dios
IXA NLP Group, University of the Basque Country (UPV/EHU)
Manuel Lardizabal 1 48014 Donostia
maxux.aranzabe@ehu.es, a.diazdeilarraza@ehu.es, igonzalez010@ikasle.ehu.es

Resumen: En este artículo se presenta uno de los módulos que forma parte del sistema de simplificación automática de textos escritos en euskera que se está implementando. Concretamente, se describe el módulo donde se lleva a cabo la transformación de las oraciones compuestas en oraciones simples. Esta transformación se realiza mediante las herramientas de alta precisión y cobertura general desarrolladas para el tratamiento automático del euskera. Además de adaptar y enriquecer el identificador de oraciones se ha implementado un algoritmo basado en árboles de dependencias sintácticas cuyo objetivo es dividir las oraciones complejas en oraciones más simples.
Palabras clave: Simplificación automática de textos, división de oraciones, euskera, identificación de las oraciones compuestas y simples

Abstract: In this paper we present a module of the Text Simplification architecture that we are implementing. Exactly, we describe the module that carries out the task of splitting sentences into clauses. This module is based on general-coverage tools. We have adapted the clause identifier in this module and we have added a algorithm based on dependency-trees to split the sentences. This way, we get simple sentences.
Keywords: Text Simplification, sentence splitting, Basque, clause boundary identification

1 *Introduction*

Automatic Text Simplification (TS) is a Natural Language Processing (NLP) task that aims the transformation of difficult texts to get a equivalent simple text. This may involve simplifyng syntactic phenomena, performing operations like sentence spliting, changing passive to active voice, inverting the order of the clauses, changing discourse marker by a simpler and/or more frequent one. As a result, this new text should easier to understand for humans and/or easier to process by NLP advanced applications and it should keep the meaning of original text, or at least information loss should be avoid.

TS systems and architectures have been proposed for languages like English (Siddharthan, 2006), Brazilian Portuguese (Candido et al., 2009), Swedish (Rybing, Smith, and Silvervarg, 2010), Japanese (Inui et al., 2003), Arabic (Al-Subaihin and Al-Khalifa, 2011), Spanish (Saggion et al., 2011), and French (Seretan, 2012). As method, dependency trees have been used in TS systems like (Zhu, Bernhard, and Gurevych, 2010) and (Siddharthan, 2011) among others.

The target audiences of the TS systems have been people with disabilities (Carroll et al., 1999), illiterate (Candido et al., 2009) or people who learn foreign languages (Petersen and Ostendorf, 2007) (Burstein, 2009) among others. There are TS system for NLP advanced applications such us machine translation (Poornima et al., 2011), Q&A systems (Bernhard et al., 2012), information extraction system (Jonnalagadda and Gonzalez, 2010), and so on.

One of the operations in TS is sentence splitting. In fact, it is a compulsory need to find precise splitting points in order to continue the next operations in the TS task. In this study we analyse two linguistic diverse structures in Basque like relative clauses and adverbial temporal clauses in other to evaluate how accurate our tools are. Besides, we implement and algorithm to create simple

ISSN 1135-5948

sentences out of a complex one. Although we get simple sentences, the simplification process is not achieved: complementisers and suffixes should be removed in other to get grammatically correct sentences.

This paper is strucutured as follows: In section 2 we describe the phenomena we have treated in this paper, namely relative clauses (subsection 2.1) and temporal adverbial clauses (subsection 2.2). In section 3 we describe the simplification process we follow together with our system architecture. In section 4 we explain how we transform the trees. After that in section 5 we present the evaluation. The conclusion and future work are presented in section 6.

2 Treated Phenomena

In order to make a deep analysis of the clause boundary identifier implemented in the splitting module we explain the two phenomena we have focused on: relative clauses and adverbial temporal clauses. We selected relative clauses since they are attached to a noun and on the other hand, adverbial temporal clauses have been chosen because they show varied structures.

The corpus that has been used for this task has been EPEC (*Euskararen Prozesamendurako Erreferentzia Corpusa*-Reference Corpus for the Processing of Basque). EPEC corpus contains 300,000 words written in Standard Basque and it is tagged at morphological and syntactical levels (dependency-trees) (Aduriz et al., 2006a). At semantic level the most frequent nouns have been tagged with their corresponding synset in EusWordNet and EusSemcor (Agirre et al., 2006). Besides, the instances of the most frequent verbs have been tagged with their thematic roles in (Aldezabal et al., 2010). At the pragmatic level, discourse markers (Iruskieta, Díaz de Ilarraza, and Lersundi, 2011) and coreference (Soraluze et al., 2012) are also tagged.

We will see in next sections examples illustrating the treated phenomena. We will only show the relevant morphological information in the glosses.

2.1 Relative clauses

Basque uses gapping as strategy for relativisation, which is marked as PRO[1]. Basque relative clause can be built with finite verbs (1) using the complementiser (comp) *-(e)n* and with non finite verbs (2), attaching to the participle the suffixes *(-ta/da, -ik, -i)* + *-ko* (rel). Let us see some examples where the relative clause is marked between brackets in the examples.

(1) *Horixe zen (magoak eta nik*
That was magician and I
genuen) sekretua.
had-COMP secret.

'That was the secret the magician and me shared.'

(2) *(Bildutako diruarekin,)*
CollectREL money-SOZ,
Afganistanerako hegazkin-txartela
Afghanistan-ALL plane-ticket
erosi zitzaion Pepitari.
buy aux Pepita-DAT

'With the collected money, a plane-ticket to Afghanistan was bought to Pepita.'

The location of finite relative clauses and non finite verb relative clauses within the sentence is at the left side of the antecedent. The subordinate verb is at the end of the relative sentence.

2.2 Adverbial temporal clauses

Adverbial temporal clauses are adjuncts that specify chronological ordering (anteriority, posteriority, simultaneity, delimitation, impendency and duration) having the reference of a main verb/clause. Temporal clauses constitute a heterogeneous group, not only semantically but syntactically too. They can be built with finite verbs and non finite verbs. In both cases free elements can be added.

Finite verb temporal clauses are headed by complementisers and suffixes are attached to verb (V) like *zu#-(e)n*V.COMP *#an*-INE in example (3). In some cases like (4) a free element (*bitartean*) is added after the verb with the complementiser. Let us see these examples, where the temporal clause is marked between brackets.

[1] Phonetically null but syntactically active element

(3) *(Jontxu ikusi zuenean,) laster*
Jontxu see aux-COMP.INE, soon
ezagutu zuen.
recognise aux
'When s/he saw Jontxu, s/he recognised him soon.'

(4) *(Indarrean egon den*
force-INE be aux-COMP
bitartean) ez du mugapenik
meanwhile not aux delimitation
izan
be
'While it has been in force, it had no delimitation.'

Non finite verb temporal clauses are formed on the basis of the verbal noun (VN) or participle. After that suffixes are added like the inessive (INE) in *itzultze*-VN#*an*-INE from (5) example. Free elements like *ostean* in (6) can be added after the verb.

(5) *(Etxera itzultzean,)*
Home-ALL come_back-INE,
Annikak makinaz pasatzen
Annika-ERG machine-INS pass
zuen testua.
aux text-ABS
'At coming back home, Annika used to type the text'.

(6) *(Maistrak agindutakoa egin*
Teacher-ERG order-REL.ABS do
ostean,) arratsalde osoa zeukaten
after, afternoon whole had
jolasteko (...)
play-FINAL CLAUSE
'After having done what the teacher ordered, they had all the afternoon to play.'

Contrary to relative clauses, the subordinate verb does not need to be always in the last position, so we can find arguments or adjuncts after it. This canonical word order alteration is difficult too for a rule based chunker, above all if there are more than one element after the verb and no punctuation marks, that could help us by giving a clue.

3 Simplification process and system architecture

In this section we present the simplification process we follow and the architecture of the system (see figure 1) we are implementing to perform the simplification process.

The simplification process illustrates the operations that should be done and the steps we follow in order to produce simple sentences out of long sentences. Before this process is initiated, the readability of the text is analysed. This task is performed by *Idazlanen Autoebaloaziorako Sistema (IAS)*[2] module (Castro-Castro et al., 2008), a system already developed by our group for the auto-evaluation of essays, which discriminates the texts that should continue the process.

Having as input a complex text, following operations are performed:

1. **Splitting:** Make as many new sentences as clauses out of the original. This operation is performed by *Mugak* (Arrieta, 2010).
2. **Reconstruction:** Two operations take place in the split sentences:
   (a) Removing no longer needed morphological features like complementisers and suffixes. Being Basque an agglutinative language we have to remove parts of words and not a whole word.
   (b) Adding new elements like adverbs or paraphrases. The goal is to maintain the meaning. In other words, the features that have been deleted should be replaced by new words. This is included in DAR (Deletion and Addition Rules) module.
3. **Reordering:** Reorder the elements in the new sentences, and ordering the sentences in the text. The set of these rules is included in ReordR (Reordering Rules) module.
4. **Adequation and Correction:** Correct the possible grammar and spelling mistakes, and fix punctuation and capitalisation. The spell checker for Basque Xuxen (Agirre et al., 1992) will carry put this operation.

[2] System for auto-evaluation of essays

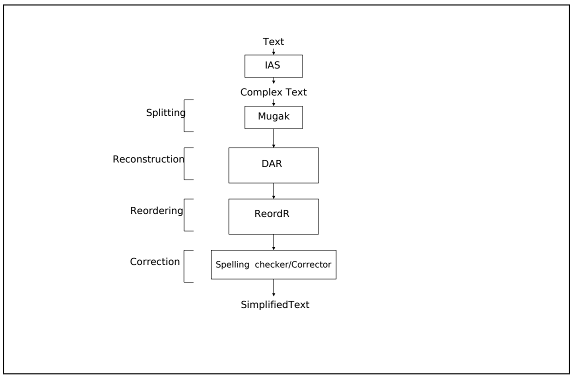

Figure 1: The architecture of system

The work presented in this paper is framed in the splitting operation and at the same time it guides the sentences to the reconstruction operation.

4 Transformation of complex sentences

Our splitting module is based on two stages: first, we apply a grammar that tags the splitting point, that is, the clause boundary is marked, and secondly, we apply an algorithm to make dependency-trees of the clauses out of the original sentence.

4.1 Splitting Point Tagging

The task of splitting point tagging is made by *Mugak* following the Constraint Grammar (CG) (Karlsson et al., 1995) formalism.

Mugak works on the basis of the output produced by several tools implemented in our group: Morpho-syntactic analysis by *Morpheus* (Alegria et al., 2002), lemmatisation and syntactic function identification by *Eustagger* (Aduriz et al., 2003), multi-words items identification (Ezeiza, 2002) (Urizar, 2012) and named entity recognition by *Eihera* (Alegria et al., 2003).

Our work consists on improving the grammar in *Mugak* (Ondarra, 2003) (Aduriz et al., 2006b) by means of adding new rules and adapting older rules based on linguistic knowledge, that lead us to get better results.

In this moment there are 78 rules and 22 of them are especially written for the phenomena we are presenting in this paper. Major improvements have been made this time in the detection of clauses headed by compound verbs and the comma. We have to remark that this is an ongoing work, that is optimised by using new corpora to find new structures and above all to determine the precision in case of non canonical order sentences.

4.2 Splitting algorithm

We have implemented an algorithm to apply several heuristics defined to transform a complex sentence into simple sentences, once the splitting point has been tagged. The usage of this algorithm is to create the dependency-trees of the new sentences. To create this algorithm and to help the following reconstruction step, we have carried out an experiment with sentences in EPEC-DEP (Basque Dependency Treebank) (Aranzabe, 2008) that were sintactically deep tagged, that is PRO[3] and pro[4] elements had a tag.

Let us explain this process by means of an example. Figure 2 shows the tree of the original sentence *Bere zeregina zatituta dagoen alderdia batzea izango dela esan zuen* (S/he said that her/his mission is to unify the political party that is divided).

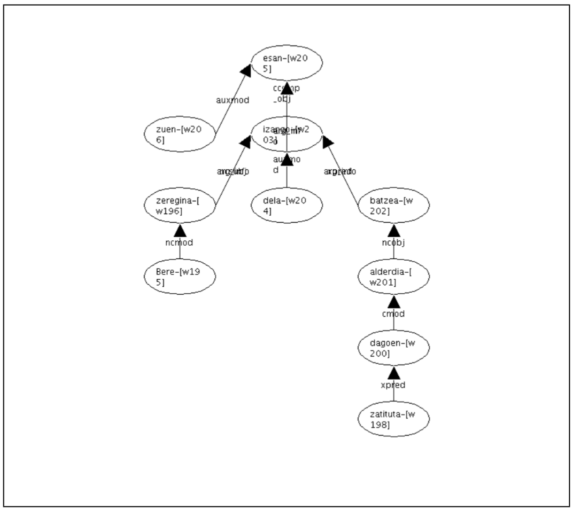

Figure 2: Original sentence: *Bere zeregina zatituta dagoen alderdia batzea izango dela esan zuen*

Having this input our algorithm works as follows:

1. The relative clause *zatituta dagoen* (that is divided) is removed out of the original sentence. This way we get two trees: the main clause *Bere zeregina alderdia batzea izango dela esan zuen.* (S/he said that her/his mission is to unify the political party.) (figure 3) and the relative

[3]see footnote 1

[4]elided arguments (pro-drop)

clause *zatituta dagoen* (that is divided) (figure 4).

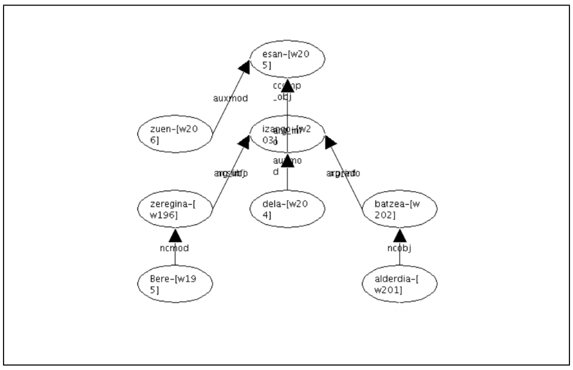

Figure 3: The main clause: *Bere zeregina alderdia batzea izango dela esan zuen*

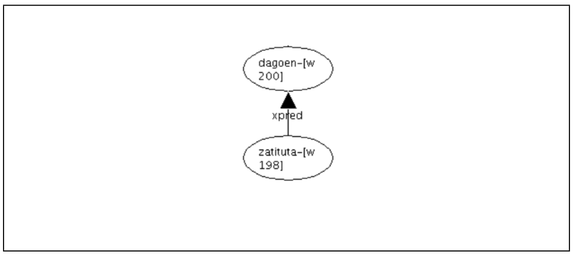

Figure 4: The relative clause: *zatituta dagoen*

2. The PRO antecedent of relative clause *alderdia* (The political party) is included in the new sentence. This way, the sentence *alderdia zatituta dagoen* is formed as shown in the tree of figure 5.

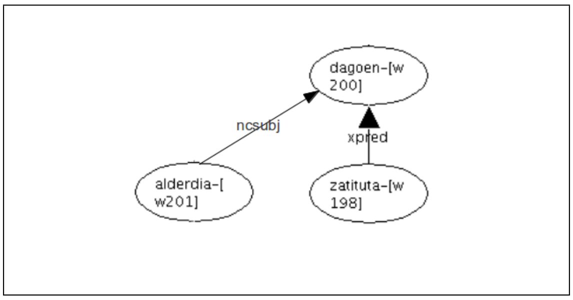

Figure 5: The new simple sentence (relative + antecedent): *alderdia zatituta dagoen*

In the case of adverbial temporal clauses, the adverbial clause is removed in the first step and an adverb will be added in the second step.

This way the reconstruction operation is over in both cases at tree level. That is simple sentences are formed, but they are not grammatically correct. The reconstruction will be over, continuing with this example, by removing the *-(e)n* complementiser of the verb.

5 Evaluation

In this section we evaluate the correctness assessing the splitting point tag and splitting the sentences.

The corpus that has been used to develop and to evaluate the grammar has been EPEC. We divided the corpus in two sets: devel and eval. We used devel for designing the rules of the grammar and eval for automatic evaluation. The latter was previously manually tagged. In table 1 we see the word and sentence number we have used for this task in the development part and the evaluation part of the corpus.

	Devel	**Eval**
Word number	61121	63766
Sentence number	5068	5211
Clause number	18301	18356

Table 1: Word, sentence and clause number in corpus

In table 2 we show the results we obtained by relative clauses and adverbial temporal clauses. The measures that we have used are precision (correctly detected clauses/detected clauses), recall (correctly detected clauses/all clauses) and F-measure (2 * precision * recall / (precision + recall)). Forth column shows the clause number of each structure.

For relative clauses, the results are high. The F-measure for the finite verbs is 0,988 and for the non finite verbs it is 0,992. By analysing the errors the chunker made we concluded that:

- We have a problem with a rule that aims a finite verb temporal clause with free elements structure that can be mixed with relative sentences.
- Another kind of error was due to errors in the PoS tagging.
- Non finite modal verbs structures were not found in the development part.

For temporal clauses, we have to divide the results in two groups: clauses without free elements and clauses with free ele-

	Precision	Recall	F-measure	Clause number
Relative finite verb clauses	0,998	0,978	0,988	547
Relative non finite verb clauses	1	0,985	0,992	335
Temporal finite verb clauses	0,955	0,964	0,960	111
Temporal non finite verb clauses	0,966	0,966	0,966	29
Temporal finite verb clauses + free element(s)	1	0,556	0,714	18
Temporal non finite verb clauses + free element(s)	0,970	0,372	0,538	86

Table 2: Evaluation results of the treated phenomena

ments. The results for the first group are quite high and similar for finite and non finite verbs. The F-measure for temporal finite verb clauses is 0,960 and for the non temporal finite verb clauses is 0,966. We analysed the errors and they are due to canonical word order alteration.

The results for the second group are, however, lower. The F-measure for the temporal finite verb clauses + free element(s) is 0,714 and for the temporal non finite verb clauses + free element(s) is 0,538. The main problem here is that the recall is very low (finite verbs 0,556 and non finite verbs 0,372). Those results are due to:

- The ambiguity of the free elements
- The richness of those structures (all of them were not found in the development part)

Anyway, apart from the problem of the ambiguity the precision we get is high (finite verbs 1 and non finite verbs 0,970).

Since our aim consists on getting accuracy (precision) it is widely achieved, so we consider that we have a basis to continue with the simplification process. This basis is extremely remarkable for relative clauses. The results of the temporal clauses are good. Nevertheless, we should keep on improving the rules, and if possible, getting more structures. It is remarkable too that recall goes down resounding when the clause has free elements, since it is difficult to cover all the possible structures with a corpus. So, defining the clause boundaries is a continuous task we have to keep on working on in order to improve our clause boundary identifier.

6 Conclusion and Future work

In this paper we have focused on the splitting module in our text simplification architecture, since we think that it is important to have a good basis to continue with the simplification process. As we have explained, this module works on two phases: clause boundary detection and splitting point tagging and building simple sentence dependency-trees out of original sentence. The first phase tagging is made by means of *Mugak* a linguistic knowledge based grammar written in the Constraint Grammar formalism and the second phase is carried out by an algorithm based on dependencies-trees as well to create so many sentences out of the clauses in the original sentences. Furthermore, this algorithm introduces the clause in the reconstruction operation.

For this task, we have deeply analysed two diverse structures, namely relative clauses and adverbial temporal clauses. We have explained their different formation and the challenge they suppose.

We have made an evaluation and concluded that we have great basis to continue with the simplification process. Moreover, the algorithm we have implemented introduces the clauses in the reconstruction step fulfilling almost the simplification process in the case of relative sentences. But, on the other hand, the improvements made here to the clause boundary identifier will serve to improve the performance of other tools which use older versions of this identifier, for example, the statistical clause boundary identifier (Arrieta, 2010).

Our next step is actually to keep on working with the syntactic simplification process. For the verb state changing, that is becoming a subordinate verb into a main verb, we plan to use finite state technology tools like FOMA (Hulden, 2009). This tool will be useful as well to implement deletion and addition rules so far defined in (Gonzalez-Dios, 2011).

Acknowledgements

Itziar Gonzalez-Dios's work is funded by a PhD grant from the Basque Government. This research was supported by the Basque Government (IT344-10), and the

Spanish Ministry of Science and Innovation (MICINN, TIN2010-20218). We thank Iñigo Lopez-Gazpio for the support in programming task.

References

Aduriz, Itziar, Izaskun Aldezabal, Iñaki Alegria, Jose Mari Arriola, Arantza Díaz de Ilarraza, Nerea Ezeiza, and Koldo Gojenola. 2003. Finite State Applications for Basque. In *EACL'2003 Workshop on Finite-State Methods in Natural Language Processing.*, pages 3–11.

Aduriz, Itziar, María Jesús Aranzabe, Jose Mari Arriola, Aitziber Atutxa, Arantza Díaz de Ilarraza, Nerea Ezeiza, Koldo Gojenola, Maite Oronoz, Aitor Soroa, and Ruben Urizar. 2006a. Methodology and steps towards the construction of EPEC, a corpus of written Basque tagged at morphological and syntactic levels for automatic processing. *Language and Computers*, 56(1):1–15.

Aduriz, Itziar, Bertol Arrieta, Jose Mari Arriola, Arantza Díaz de Ilarraza, Elixabete Izagirre, and Ainara Ondarra. 2006b. Muga Gramatikaren Optimizazioa. Technical report, UPV/EHU/LSI/TR 9-2006.

Agirre, Eneko, Izaskun Aldezabal, Jone Etxeberria, Mikel Iruskieta, Elixabete Izagirre, Karmele Mendizabal, and Eli Pociello. 2006. A methodology for the joint development of the Basque WordNet and Semcor. In *Proceedings of the 5th International Conference on Language Resources and Evaluations (LREC).*

Agirre, Eneko, Iñaki Alegria, Xabier Arregi, Xabier Artola, Arantza Díaz de Ilarraza, Montse Maritxalar, Kepa Sarasola, and Miriam Urkia. 1992. Xuxen: A Spelling Checker/Corrector for Basque based in Two-Level Morphology. In *Proceedings of NAACL-ANLP'92*, pages 119–125.

Al-Subaihin, Afnan A. and Hend S. Al-Khalifa. 2011. Al-Baseet: A proposed simplification authoring tool for the Arabic language. In *International Conference on Communications and Information Technology (ICCIT)*, pages 121–125.

Aldezabal, Izaskun, María Jesús Aranzabe, Arantza Díaz de Ilarraza, Ainara Estarrona, Kike Fernandez, and Larraitz Uria. 2010. EPEC-RS: EPEC (Euskararen Prozesamendurako Erreferentzia Corpusa) rol semantikoekin etiketatzeko eskuliburua. Technical report, UPV/EHU/LSI/TR 02-2010.

Alegria, Iñaki, María Jesús Aranzabe, Aitzol Ezeiza, Nerea Ezeiza, and Ruben Urizar. 2002. Robustness and customisation in an analyser/lemmatiser for Basque. In *LREC-2002 Customizing knowledge in NLP applications workshop*, pages 1–6.

Alegria, Iñaki, Nerea Ezeiza, Izaskun Fernandez, and Ruben Urizar. 2003. Named Entity Recognition and Classification for texts in Basque. In *II Jornadas de Tratamiento y Recuperación de Información, JOTRI.*

Aranzabe, María Jesús. 2008. *Dependentzia-ereduan oinarritutako baliabide sintaktikoak: zuhaitz-bankua eta gramatika konputazionala.* Ph.D. thesis, Euskal Filologia Saila (UPV/EHU).

Arrieta, Bertol. 2010. *Azaleko sintaxiaren tratamendua ikasketa automatikoko tekniken bidez: euskarako kateen eta perpausen identifikazioa eta bere erabilera koma-zuzentzaile batean.* Ph.D. thesis, Informatika Fakultatea (UPV-EHU).

Bernhard, Delphine, Louis De Viron, Véronique Moriceau, and Xavier Tannier. 2012. Question Generation for French: Collating Parsers and Paraphrasing Questions. *Dialogue and Discourse*, 3(2):43–74.

Burstein, Jill. 2009. Opportunities for Natural Language Processing Research in Education. In *Computational Linguistics and Intelligent Text Processing.* Springer Berlin / Heidelberg.

Candido, Jr., Arnaldo, Erick Maziero, Caroline Gasperin, Thiago A. S. Pardo, Lucia Specia, and Sandra M. Aluisio. 2009. Supporting the adaptation of texts for poor literacy readers: a text simplification editor for Brazilian Portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, EdAppsNLP '09, pages 34–42. ACL.

Carroll, John, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying Text for

Language-Impaired Readers. In *9th Conference of the European Chapter of the Association for Computational Linguistics*.

Castro-Castro, Daniel, Rocío Lannes-Losada, Montse Maritxalar, Ianire Niebla, Celia Pérez-Marqués, Nancy C. Alamo-Suarez, and Aurora Pons-Porrata. 2008. A Multilingual Application for Automated Essay Scoring. In *Lecture Notes in Advances in Artificial Intelligence - LNAI 5290 - IBERAMIA*, pages 243–251. Springer New York.

Ezeiza, Nerea. 2002. *Corpusak ustiatzeko tresna linguistikoak. Euskararen etiketatzaile morfosintaktiko sendo eta malgua.* Ph.D. thesis, Informatika Fakultatea, UPV-EHU.

Gonzalez-Dios, Itziar. 2011. Euskarazko egitura sintaktikoen azterketa testuen sinplifikazio automatikorako: Aposizioak, erlatibozko perpausak eta denborazko perpausak. Master's thesis, University of the Basque Country.

Hulden, Mans. 2009. Foma: a Finite-State Compiler and Library. In *EACL (Demos)'09*, pages 29–32.

Inui, Kentaro, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: a project note. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, pages 9–16. ACL.

Iruskieta, Mikel, Arantza Díaz de Ilarraza, and Mikel Lersundi. 2011. Unidad discursiva y relaciones retóricas: un estudio acerca de las unidades de discurso en el etiquetado de un corpus en euskera. *Procesamiento del Lenguaje Natural*, (47).

Jonnalagadda, Siddhartha and Graciela Gonzalez. 2010. Sentence simplification aids protein-protein interaction extraction. *Arxiv preprint arXiv:1001.4273*.

Karlsson, Fred, Atro Voutilainen, Juha Heikkila, and Atro Anttila. 1995. *Constraint Grammar, A Language-independent System for Parsing Unrestricted Text.* Mouton de Gruyter.

Ondarra, Ainara. 2003. Murriztapen Gramatikaren sintaxia. EUSMG optimizatzen. Esaldi-mugak. Master's thesis, Euskal Herriko Unibertsitatea.

Petersen, Sarah E. and Mari Ostendorf. 2007. Text Simplification for Language Learners: A Corpus Analysis. *Electrical Engineering*, (SLaTE):69–72.

Poornima, C., V. Dhanalakshmi, K.M. Anand, and KP Soman. 2011. Rule based Sentence Simplification for English to Tamil Machine Translation System. *International Journal of Computer Applications*, 25(8):38–42.

Rybing, Jonas, Christian Smith, and Annika Silvervarg. 2010. Towards a Rule Based System for Automatic Simplification of texts. In *The Third Swedish Language Technology Conference (SLTC 2010)*.

Saggion, Horacio, Elena Gómez-Martínez, Esteban Etayo, Alberto Anula, and Lorena Bourg. 2011. Text Simplification in Simplext: Making Text More Accessible. *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural*.

Seretan, Violeta. 2012. Acquisition of syntactic simplification rules for french. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA).

Siddharthan, Advaith. 2006. Syntactic simplification and text cohesion. *Research on Language & Computation*, 4(1):77–109.

Siddharthan, Advaith. 2011. Text Simplification using Typed Dependencies: A Comparison of the Robustness of Different Generation Strategies. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 2–11. ACL.

Soraluze, Ander, Olatz Arregi, Xabier Arregi, Klara Ceberio, and Arantza Díaz de Ilarraza. 2012. Mention Detection: First Steps in the Development of a Basque Coreference Resolution System. In *Proceedings of KONVENS 2012 (Main track: oral presentations)*, pages 128–163.

Urizar, Ruben. 2012. *Euskal lokuzioen tratamendu konputazionala.* Ph.D. thesis, UPV-EHU.

Zhu, Zhemin, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of The 23rd International Conference on Computational Linguistics*, pages 1353–1361.