Interactive Predictive Parsing Framework for the Spanish Language

Marco para Parsing Predictivo Interactivo aplicado a la Lengua Castellana

Ricardo Sánchez-Sáez, Luis A. Leiva, Joan Andreu Sánchez, and José Miguel Benedí

Resumen: El marco teórico de Parsing Predictivo Interactivo (IPP) permite construir sistemas de anotación sintáctica interactivos. Los anotadores humanos pueden utilizar estos sistemas de ayuda para crear árboles sintácticos con muy poco esfuerzo (en comparación con el trabajo requerido para corregir manualmente árboles obtenidos a partir de un analizador sintáctico completamente automático). En este artículo se presenta la adaptación a la lengua castellana del marco IPP y su herramienta de anotación *IPP-Ann*, usando modelos obtenidos a partir del UAM Spanish Treebank. Hemos llevado a cabo experimentación simulando al usuario para obtener métricas de evaluación objetivas para nuestro sistema. Estos resultados muestran que el marco IPP aplicado al UAM Spanish Treebank se traduce en una importante cantidad de esfuerzo ahorrado, comparable con el obtenido al aplicar el marco IPP para analizar la lengua inglesa mediante el Penn Treebank.
Palabras clave: Parsing, Parsing Predictivo Interactivo, lengua castellana

Abstract: The Interactive Predictive Parsing (IPP) framework allows the construction of interactive tree annotation systems. These can help human annotators in creating error-free parse trees with little effort (compared to manually post-editing the trees obtained from a completely automatic parser). In this paper we adapt the IPP framework and the *IPP-Ann* annotation tool for parse of the Spanish language, by using models obtained from the UAM Spanish Treebank. We performed user simulation experimentation and obtained objective evaluation metrics. The results establish that the IPP framework over the UAM Treebank shows important amounts of user effort reduction, comparable to the gains obtained when applying IPP to the English language on the Penn Treebank.
Keywords: Parsing, Interactive Predictive Parsing, Spanish language

1 Introduction

Two different usage cases can be acknowledged for automatic systems that output, or work with, natural language within the Computational Linguistics field. On one hand, we have the scenario in which the output of such systems is expected to be used in a vanilla fashion, that is, without validating or correcting the results produced by the system. Within this usage scheme, the most important factor of a given automatic system is the quality of the results. Although memory and computational requirements of such systems are usually taken into account, the ultimate aim of most research that relates to this scenario is to minimize the amount of error (measured through metrics like Word Error Rate, BLEU, F-Measure, etc.) inherent to the produced results.

A second usage scene arises when there exists the need for perfect and completely error-free results, e.g., perfectly translated sentences or perfectly annotated syntactic trees. In such a case, the intervention of a human user validator/corrector is unavoidable. The corrector will review and validate the results, making the suitable corrections before the system output can be employed. In these kind of tasks, the most important factor that has to be minimized is the human effort that has to be applied to transform system's potentially incorrect output into validated and error-free output. Measuring user effort has

an intrinsic subjectivity that makes it hard to be quantitatized. Most research about problems associated to this scenario tries to minimize just the system's error rate as well, given the fact that user effort is usually inversely proportional to the quality of the output.

Only recently, more comparable and reproducible evaluation methods for Interactive Natural Language Systems have started to be developed within the context of Interactive Predictive Systems. These systems formally integrate the correcting user into the loop, making him part of the system, right at its theoretical framework. Interactive predictive methods have been studied and successfully used in fields like Handwritten Text Recognition (HTR) (Toselli, Romero, and Vidal, 2008; Romero et al., 2009) and Statistical Machine Translation (SMT) (Ortiz et al., 2010; Alabau et al., 2009) to ease the work of transcriptors and translators, respectively.

In such systems, the importance of the base system error rate *per se* is diminished. Instead, the intention is to measure how well the user and the system work together. For this, formal user simulation protocols together with new objective effort evaluation metrics such as the Word Stroke Ratio (WSR) (Toselli, Romero, and Vidal, 2008) or the Key-Stroke and Mouse-Ratio (KSMR) (Barrachina et al., 2009) started to be used as a benchmark. These ratios reflect the amount of user effort (whole-word corrections in the case of WSR; keystrokes plus mouse actions in the case of KSMR) given a certain output. To get the amount of user effort into context they should be compared against the corresponding error ratios of comparable non-interactive systems: Word Error Rate in the case of WSR and Character Error Rate in the case of KSMR.

This dichotomy in evaluating either system performance or user effort applies to Syntactic Parsing as well. The objective of parsing is to precisely determine the syntactic structure of sentences written in one of the several languages that humans use. Some examples of top performing completely automatic parsers are (Collins, 2003; Klein and Manning, 2003; Petrov and Klein, 2007; McClosky, Charniak, and Johnson, 2006; Huang, 2008).

In the parsing field, there exist a dire need for manually annotated corpora are needed, specially for languages in which parse corpora are sparse. Annotating trees syntactically generally requires human intervention of a high degree of specialization. This fact partially justifies the shortage in large manually annotated treebanks. Endeavors directed at easing the burden for the experts performing this task could be of great help, such as the ones presented in (de la Clergerie et al., 2008).

When using automatic parsers as a baseline for building perfect syntactic trees, the role of the human annotator is usually to post-edit the trees and correct the errors. This manner of operation results in the typical two-step process for error correcting, in which the system first generates the whole output and the user verifies or amends it. This paradigm is rather inefficient and uncomfortable for the human annotator. For example, a basic two-stage setup was employed in the creation of the Penn Treebank annotated corpus: a rudimentary parsing system provided a skeletal syntactic representation, which then was manually corrected by human annotators (Marcus, Santorini, and Marcinkiewicz, 1994). Additional works within this field have presented systems that act as a computerized aid to the user in obtaining the perfect annotation (Carter, 1997; Oepen et al., 2004; Hiroshi et al., 2005). Subjective measuring of the effort needed to obtain perfect annotations was reported in some of these works, but we feel that a more comparable metric is needed.

With the objective of reducing the user effort and making the laborious task of tree annotation easier, the authors of (Sánchez-Sáez, Sánchez, and Benedí, 2009) devised an Interactive Predictive Parsing framework. That work embeds the human corrector into the automatic parser, and allows him to interact in real time within the system. In this manner, the system can use the readily available user feedback to make predictions about the parts of the trees that have not been validated by the corrector. The authors performed experiments over the Penn Treebank: they simulated user interaction and calculated effort evaluation metrics, establishing that an IPP system results in amounts slightly above 40% of effort reduction for a manual annotator compared to a two-step system. In (Sánchez-Sáez et al., 2010) they also demonstrated the Interactive Predictive Parsing Tree Annotator (IPP-Ann) an IPP

based annotation tool that can be accessed at `http://cat.iti.upv.es/ipp/`.

In this paper, we apply the IPP framework to the Spanish language, by updating its model to Probabilistic Context Free Grammars (PCFGs) obtained from the UAM Spanish Treebank (Moreno et al., 2000). We also adapted *IPP-Ann* to parse sentences in the Spanish language, which could pave the way to further developments in order to make this tool compatible with other annotation styles. *IPP-Ann*, by helping to syntactically annotate new sentences more efficiently, could be a very helpful asset in increasing the size of the UAM corpus, or in the creation of other Spanish treebanks.

In order to quantitatively measure IPP performance on the Spanish language, we also carried out user simulation experimentation with the UAM Treebank to determine that effort reduction estimates for Spanish are comparable to the figures obtained for English parsing using the Penn Treebank.

2 *Interactive Predictive Parsing*

In this section we review the IPP framework (Sánchez-Sáez, Sánchez, and Benedí, 2009) and its underlying operation protocol. In parsing, a syntactic tree t, attached to a string $\boldsymbol{x} = x_1 \dots x_{|x|}$, is composed by substructures called constituents. A constituent c_{ij}^A is defined by the nonterminal symbol (either a *syntactic label* or a *POS tag*) A and its span ij (the starting and ending indexes which delimit the part of the input sentence encompassed by the constituent).

Here follows a general formulation for the non-interactive syntactic parsing scenario, which will allow us to better introduce the IPP formulation. Assume that using a given parsing model G, the parser analyzes the input sentence $\boldsymbol{x}$ and produces the most probable parse tree

$$\hat{t} = \arg\max_{t \in \mathcal{T}} p_G(t|\boldsymbol{x}), \tag{1}$$

where $p_G(t|\boldsymbol{x})$ is the probability of parse tree t given the input string $\boldsymbol{x}$ using model G, and $\mathcal{T}$ is the set of all possible parse trees for $\boldsymbol{x}$.

In the IPP framework, the manual corrector provides feedback to the system by correcting any of the constituents c_{ij}^A from $\hat{t}$. The system reacts to each of the corrections performed by the human annotator by proposing a new $\hat{t}'$ that takes into account the correction.

Within the IPP framework, the user reviews the constituents contained in the tree to assess their correctness. When the user finds an incorrect constituent he modifies it, setting the correct span and label. This action implicitly validates what it is called *the validated prefix tree* t_p, which is composed by the partially corrected constituent, all of its ancestor constituents, and all constituent whose end span is lower than the start span of the corrected constituent. When the user replaces the constituent c_{ij}^A with the correct one $c_{ij}'^A$, the validated prefix tree is

$$\begin{aligned} t_p(c_{ij}'^A) = \{ & c_{mn}^B \ : m \le i,\ n \ge j\,, \\ & d(c_{mn}^B) \le d(c_{ij}'^A)\} \cup \\ & \{c_{pq}^D \ : q < i\ \} \end{aligned} \tag{2}$$

with $d(c_{ab}^Z)$ being the depth (distance from root) of constituent c_{ab}^Z. The validated prefix tree is parallel to the validated sentence prefix commonly used in Interactive Machine Translation or Interactive Handwritten Recognition. This particular definition of the prefix tree that is validated after each user correction prefix determines the fact that the user is simulated by a tree exploration in a preorder fashion (left-to-right depth-first). It is worth noting that other types of prefixes could be defined, allowing for different parse trees review orders.

Within the IPP formulation, when a constituent correction is performed, the prefix tree $t_p(c_{ij}'^A)$ is validated and a new tree $\hat{t}'$ that takes into account the prefix is proposed. Incorporating this new evidence into expression (1) yields the following equation

$$\hat{t}' = \arg\max_{t \in \mathcal{T}} p_G(t|\boldsymbol{x}, t_p(c_{ij}'^A)). \tag{3}$$

Given the properties of context-free grammars, the only subtree that effectively needs to be recalculated is the one starting from the parent of the corrected constituent. This way, just the descendants of the newly introduced constituent, as well as its right hand siblings (along with their descendants) are calculated.

2.1 User Interaction Operation

The IPP formulation allows for a very straightforward operation protocol that is

performed by the manual corrector, in which he validates or corrects the successive output parse trees:

1. The parsing system proposes a full parse tree t for the input sentence.
2. Then, the user finds the first incorrect constituent exploring the tree in a certain ordered manner (preorder in our case, given by the tree prefix definition) and amends it, by modifying its span and/or label (implicitly validating the prefix tree t_p).
3. The parsing system produces the most probable tree that is compatible with the validated prefix tree t_p, as shown in expression 3.
4. These steps are iterated until a final, perfect parse tree is produced by the server and validated by the user.

It is worth noting that within this protocol, constituents can be automatically deleted or inserted by adequately modifying the span of the left-neighbouring constituent.

The IPP interaction process is similar to the ones already established in HTR and SMT. In these fields, the user reads the output sentence from left to right. When the user finds and corrects an erroneous word, he is implicitly validating the prefix sentence up to that word. The remaining suffix sentence is recalculated by the system taking into account the validated prefix sentence.

2.2 IPP-Ann

The Interactive Predictive Parsing Tree Annotator (Sánchez-Sáez et al., 2010), or *IPP-Ann* for short, is a Web-based tool based on the IPP framework. It consists on a thin Web client that operates in conjunction with a parse server which provides the parse candidates. A preview version can be accessed at `http://cat.iti.upv.es/ipp/`.

When using IPP-Ann, the user is presented with the sentences from the selected corpus, and can start parsing them one by one. The user, following the operation protocol introduced in Section 2.1, makes corrections in the trees using the keyboard and the mouse. The user feedback is decoded on the client side which in turn requests subtrees to the parse engine.

Two kind of operations can be performed over constituents: span modification (performed by dragging a line from the constituent to the word that corresponds to the span's upper index), and label substitution (done by typing the correct label on the corresponding text field). Modifying the span of a constituent invalidates its label, so the server recalculates it as part of the suffix. Modifying the label of a constituent validates its span.

As already mentioned, constituents can be adequately inserted or deleted by modifying the span of their left-neighbouring constituents. Also, an operation for inserting unary productions is available (performed by dragging a line from the parent constituent to the floating ball). Unary productions can be deleted by resetting the current span of the parent constituent.

Figure 1 shows an example of a user interaction on the IPP system. In this example, the user reviews the output tree (Fig. 1(a)) and notices that *"defiende las fusiones"* should a Verb Phrase (VP). He increases the span of V, which originally only spanned *"defiende"*. This operation validates the prefix, which can be seen highlighted in green on the user interface (Fig. 1(b)). The IPP engine correctly calculates the most suitable label for the new span, and recalculates the most probable suffix compatible with the validated prefix (Fig. 1(c)).

When the user is about to perform an operation, the affected constituent and the prefix that will be validated are highlighted. The target span of the modified constituent is visually shown as well. When the user obtains the correctly annotated tree, he can accept it by clicking on a new sentence.

3 Evaluation of Interactive Parsing Systems

As already mentioned, the objective of IPP parsing is to be employed by annotators to construct correct syntactic trees with less effort. The metrics presented here evaluate the amount of effort (consisting in the amount of constituent corrections performed using the IPP system) saved by the user, compared to the effort required to manually post-edit the trees after obtaining them with an automatic parsing system (consisting in the amount of incorrect constituents).

It is subjective and expensive to test an in-

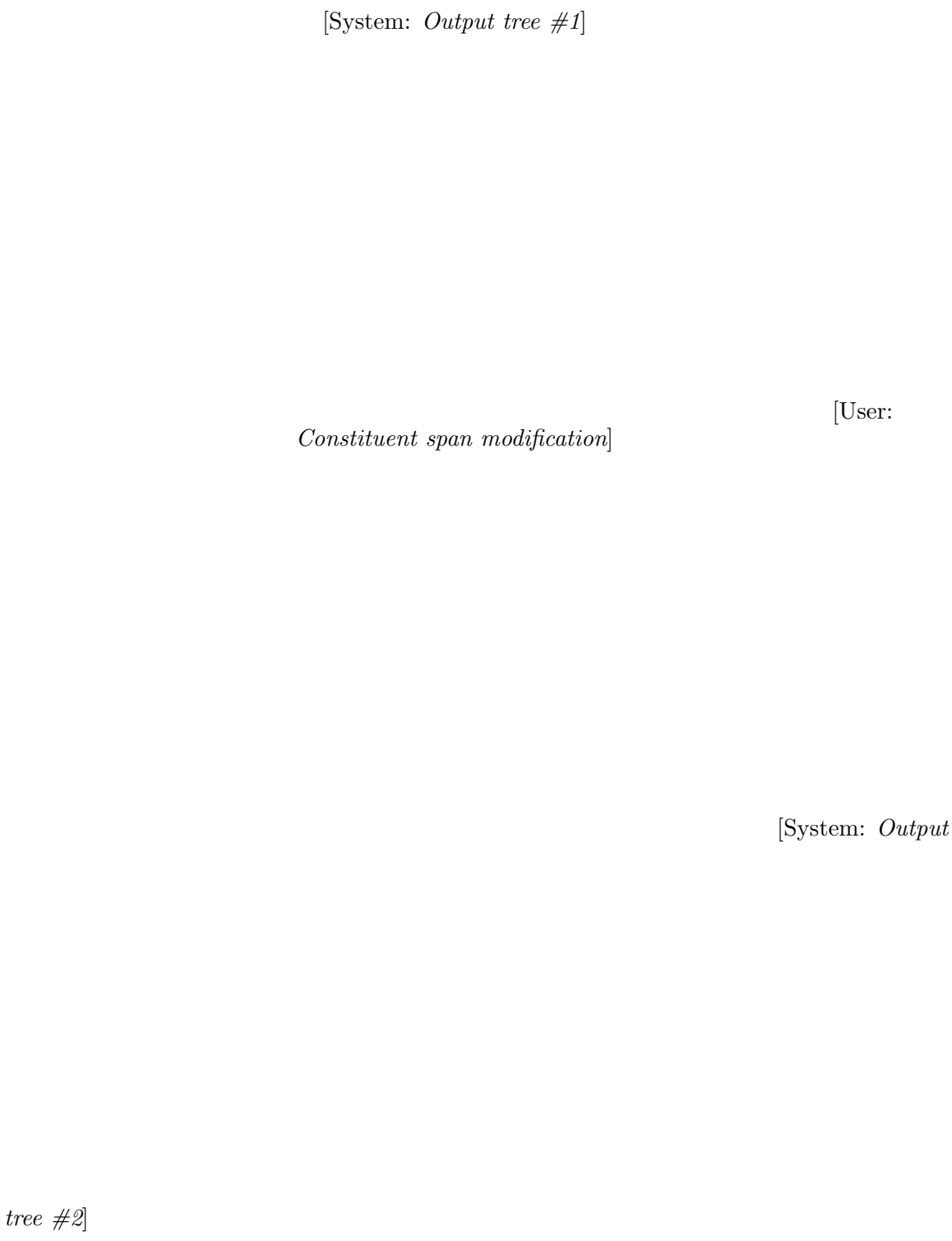

Figure 1: A user interaction on *IPP-Ann*.

teractive system with real users, so the gold reference trees were used to simulate system interaction by a human corrector and provide a comparable benchmark. The evaluation protocol is as follows:

1. The IPP system proposes a full parse tree t for the input sentence.
2. The user simulation subsystem finds the first incorrect constituent by exploring the tree in the order defined by the prefix tree definition (preorder) and comparing it with the *reference*. When the first erroneous constituent is found, it is amended by being replaced in the output tree by the correct one, operation

which implicitly validates the prefix tree t_p. The number of interactions (constituent replacements) that have been performed to obtain the perfect tree is accumulated through this process.

3. The parsing server produces the most probable tree that is compatible with the validated prefix tree t_p.

4. These steps are iterated until a final, perfect parse tree is produced by the server and validated by the user simulation subsystem.

At the end of this process, two metrics can be reported. The TCER measures the amount of user effort in obtaining perfectly annotated trees by post-editing the ones that were output by a non-interactive system. The TCAC measures the user effort in obtaining these same trees by interactively using the IPP system:

- *Tree Constituent Error Rate (TCER):* Minimum number of constituent substitution, deletion and insertion operations needed to convert the first proposed parse tree into the corresponding gold reference tree, divided by the total number of constituents in the reference tree.

- *Tree Constituent Action Rate (TCAC):* Number of constituent corrections performed by the user simulation system in conjunction with the IPP system to obtain the reference tree, divided by the total number of constituents in the reference tree.

These two metrics are directly comparable because both refer to modifications at the constituent level. For our experiments we will also report the more classical *F-Measure* metric for contextualization, which is in fact inversely related to the TCER.

4 Experiments

4.1 The UAM Spanish Treebank

The UAM Spanish Treebank[1] is a mannually annotated corpus developed at the *Laboratorio de Lingüística Informática* of the *Universidad Autónoma de Madrid* (Moreno et al., 2000). Its annotation scheme is an adaptation of the Penn Treebank style to the Spanish language (syntactic labels and POS tags have been conformed to fit Spanish sentence structures and word functions), with some additional features added. The corpus consists of 1,500 annotated sentences (22,695 words) and averages 15.13 words/sentence. The sentences were taken from the Spanish newspaper *El País* and the consumer association magazine *Compra Maestra*.

Applying the IPP framework to the UAM Treebank serves a double purpose. On one hand, we adapt the framework and *IPP-Ann* to the Spanish language, opening the door for collaborations with Spanish linguists, in order to build and improve a useful and reliable tool for effortless tree annotation.

On the other hand, experiments on this corpus also allows us to study how the IPP framework fares using a considerably small grammar as the model. Previous IPP experimentation carried out by Sánchez-Sáez et al. (1999) used Penn Treebank grammars, induced from just over 39,800 sentences (about 950,000 words). For our experiments we are inducing the Spanish grammars from a much smaller treebank set comprising of 1.400 sentences (22,785 words).

4.2 Experimental Framework

In readying our experimentation, we divided the UAM Treebank in two partitions: the train set (first 1.400 sentences) and the test set (last 100 sentences). Before carrying out experiments, the *NoEmpties* transformation was applied to both sets (Klein and Manning, 2001).

We implemented the CYK-Viterbi parsing algorithm as the parse engine within the IPP framework. This algorithm uses grammars in the Chomsky Normal Form (CNF), so we employed the open source Natural Language Toolkit (NLTK) to obtain several right-factored binary grammars with different markovization parameters from the training set. (Klein and Manning, 2003).

A basic smoothing method was used for parsing sentences with out-of-vocabulary words: when an input word could not be derived by any of the preterminals in the UAM treebank grammar, a very small probability for that word was uniformly added to all of the preterminals.

User simulation was performed as an ob-

[1] `http://www.lllf.uam.es/~sandoval/UAMTreebank.html`

jective way of IPP effort reduction evaluation, as explained in Section 3. Results for the discussed metricsfor different markovizations of the train grammar are shown in Table 1.

PCFG	Baseline		IPP	RelRed
	F_1	TCER	TCAC	
h=0, v=0	0.57	0.48	0.26	46%
h=0, v=1	0.59	0.47	0.25	47%
h=0, v=2	0.62	0.44	0.24	46%
h=0, v=3	0.61	0.45	0.24	47%

Table 1: Results for the test set: F_1 and TCER for the baseline system; TCAC for the IPP system; relative reduction beteween TCER and TCAC.

Note that baseline F_1 scores are far from the *state of the art in parsing* (in part owing to the small size of the treebank used to induce the grammar, and also owing to the use of an unlexicalized parsing method). However, our purpose is to evaluate the help of an IPP system in obtaining perfectly annotated sentences, so the relative reductions in annotation effort were calculated.

We observe high amounts of effort saving when using an IPP system to annotate sentences in an error-free fashion. Metrics show that the percentage of corrections needed using the IPP system is much lower than the rate of needed corrections when post-editing the baseline trees: an estimated 46% of constituent corrections could be saved by a human linguist using IPP-Ann.

These results are comparable to those obtained in (Sánchez-Sáez, Sánchez, and Benedí, 2009) for the Penn Treebank, which ranged effort savings from 42% to 46%. We conclude that important amounts of effort reduction in annotation is obtained from IPP, even when smaller PCFGs are used for parsing.

5 Conclusions and Future Work

We have applied the IPP framework for parsing of the Spanish language, by obtaining Probabilistic Context Free Grammars (PCFGs) from the UAM Spanish Treebank (Moreno et al., 2000). By using the same PCFGs, we also adapted the *IPP-Ann* annotation tool to parse sentences in the Spanish language. We performed user simulation experiments for perfectly annotating Spanish sentences using the UAM Treebank, with an estimated effort decrement of about 46%. The amount of effort reduction is comparable to the amount of savings obtained for IPP in English language annotation (on the Penn Treebank).

Future work involves further developments of the IPP framework and *IPP-Ann* in order to make it fully compatible with additional annotation styles, so it can be used in the field for fast treebank creation.

Long term future research deals with the addition of Adaptative Parsing algorithms to the IPP framework, which would allow to improve its model with new ground truth data as the user annotates and validates new trees.

Bibliografía

Alabau, V., D. Ortiz, V. Romero, and J. Ocampo. 2009. A multimodal predictive-interactive application for computer assisted transcription and translation. In *ICMI-MLMI '09: Proceedings of the 2009 international conference on Multimodal interfaces*, pages 227–228, New York, NY, USA. ACM.

Barrachina, Sergio, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Carter, D. 1997. The TreeBanker. A tool for supervised training of parsed corpora. In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, pages 9–15.

Collins, M. 2003. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637.

de la Clergerie, E.V., O. Hamon, D. Mostefa, C. Ayache, P. Paroubek, and A. Vilnat. 2008. Passage: from French parser evaluation to large sized treebank. *LREC*, 100:P2.

Hiroshi, I., N. Masaki, H. Taiichi, T. Takenobu, and T. Hozumi. 2005. eBonsai: An integrated environment for annotating treebanks. In *Second International Joint Conference on Natural Language Processing*, pages 108–113.

Huang, L. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proc. of ACL*. Citeseer.

Klein, D. and C.D. Manning. 2001. Parsing with treebank grammars: Empirical bounds, theoretical models, and the structure of the Penn treebank. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 338–345. Association for Computational Linguistics Morristown, NJ, USA.

Klein, D. and C.D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics Morristown, NJ, USA.

Marcus, M.P., B. Santorini, and M.A. Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

McClosky, D., E. Charniak, and M. Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159.

Moreno, A., R. Grishman, S. López, F. Sánchez, and S. Sekine. 2000. A treebank of Spanish and its application to parsing. In *Proceedings of the Second International Conference on Language Resources and Evaluation LREC*, pages 107–112. Citeseer.

Oepen, S., D. Flickinger, K. Toutanova, and C.D. Manning. 2004. LinGO Redwoods. *Research on Language & Computation*, 2(4):575–596.

Ortiz, D., L.A. Leiva, V. Alabau, and F. Casacuberta. 2010. Interactive machine translation using a web-based architecture. In *Procedings of the International Conference on Intelligent User Interfaces*. Hong Kong, China, February, pages 423–425.

Petrov, S. and D. Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of NAACL HLT 2007*, pages 404–411.

Romero, V., L.A. Leiva, A.H. Toselli, and E. Vidal. 2009. Interactive multimodal transcription of text imagse using a web-based demo system. In *Procedings of the International Conference on Intelligent User Interfaces*. Sanibel Island, Florida, February, pages 477–478.

Sánchez-Sáez, R., L.A. Leiva, J.A. Sánchez, and J.M. Benedí. 2010. Interactive predictive parsing using a web-based architecture. In *Proceedings of the Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'10, system demonstration)*, Los Angeles, United States of America, June.

Sánchez-Sáez, R., J.A. Sánchez, and J.M. Benedí. 2009. Interactive predictive parsing. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 222–225, Paris, France, October. Association for Computational Linguistics.

Toselli, A.H., V. Romero, and E. Vidal. 2008. Computer assisted transcription of text images and multimodal interaction. In *Proceedings of the 5th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, volume 5237, pages 296–308. Springer.