

Interactive Predictive Parsing ¹

Ricardo Sánchez-Sáez, Joan-Andreu Sánchez and José-Miguel Benedí

Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
Camí de Vera s/n, Valencia 46022 (Spain)
{rsanchez, jandreu, jbenedi}@dsic.upv.es

Abstract

This paper introduces a formal framework that presents a novel Interactive Predictive Parsing schema which can be operated by a user, tightly integrated into the system, to obtain error free trees. This compares to the classical two-step schema of manually post-editing the erroneous constituents produced by the parsing system. We have simulated interaction and calculated evaluation metrics, which established that an IPP system results in a high amount of effort reduction for a manual annotator compared to a two-step system.

1 Introduction

The aim of parsing is to obtain the linguistic interpretation of sentences, that is, their underlying syntactic structure. This task is one of the fundamental pieces needed by a computer to understand language as used by humans, and has many applications in Natural Language Processing (Lease et al., 2006).

A wide array of parsing methods exist, including those based on Probabilistic Context-Free Grammars (PCFGs). (Charniak, 2000; Collins, 2003; Johnson, 1998; Klein and Manning, 2003; Matsuzaki et al., 2005; Petrov and Klein, 2007). The most impressive results are achieved by subtree reranking systems, as shown in the semi-supervised method of (McClosky et al., 2006), or the forest reranking approximation of (Huang, 2008) in which packed parse forests (compact structures that contain many possible tree derivations) are used.

These state-of-the-art parsers provide trees of excellent quality. However, perfect results are vir-

tually never achieved. If the need of one-hundred-percent error free trees arises, the supervision of a user that post-edits and corrects the errors is unavoidable.

Error free trees are needed in many tasks such as handwritten mathematical expressions recognition (Yamamoto et al., 2006), or creation of new gold standard treebanks (Delacgerie et al., 2008)). For example, in the creation of the Penn Treebank grammar, a basic two-stage setup was employed: a rudimentary parsing system provided a skeletal syntactic representation, which then was manually corrected by human annotators (Marcus et al., 1993).

In this paper, we introduce a new formal framework that tightly integrates the user within the parsing system itself, rather than keeping him isolated from the automatic tools used in a classical two-step approach. This approach introduces the user into the parsing system, and we will call it “interactive predictive parsing”, or simply IPP. An IPP system is interactive because the user is in continuous contact with the parsing process, sending and receiving feedback. An IPP system is also predictive because it reacts to the user corrections: it predicts and suggest new parse trees taking into account the new gold knowledge received from the user. Interactive predictive methods have been studied and successfully used in fields like Automatic Text Recognition (Toselli et al., 2008) and Statistical Machine Translation (Barrachina et al., 2009; Vidal et al., 2006) to ease the work of transcriber and translators.

Assessment of the amount of effort saved by the IPP system will be measured by automatically calculated metrics.

2 Interactive Predictive Parsing

A tree t , associated to a string $x_{1|x|}$, is composed by substructures that are usually referred as constituents or edges. A constituent c_{ij}^A is a span de-

¹Work supported by the MIPRCV “Consolider Ingenio 2010” (CSD2007-00018), iTransDoc (TIN2006-15694-CO2-01) and Prometeo (PROMETEO/2009/014) research projects, and the FPU fellowship AP2006-01363.

finied by a nonterminal symbol (or syntactic tag) A that covers the substring x_{ij} .

Assume that using a given probabilistic context-free grammar G as the model, the parser analyzes the input sentence $\mathbf{x} = x_1 \dots x_{|\mathbf{x}|}$ and produces the parse tree \hat{t}

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

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

In an interactive predictive scenario, after obtaining the (probably incorrect) best tree \hat{t} , the user is able to modify the edges c_{ij}^A that are incorrect. The system reacts to each of the corrections introduced by the human by proposing a new \hat{t}' that takes into account the corrected edge. The order in which incorrect constituents are reviewed determines the amount of effort reduction given by the degree of correctness of the subsequent proposed trees.

There exist several ways in which a human analyzes a sentence. A top-to-bottom may be considered natural way of proceeding, and we follow this approach in this work. This way, when a higher level constituent is corrected, possible erroneous constituents at lower levels are expectedly automatically recalculated.

The introduced IPP interaction process is similar to the ones already established in Computer-Assisted Text Recognition and Computer-Assisted Translation¹.

Within the IPP framework, the user reviews the constituents contained in the tree to assess their correctness. When the user find an incorrect edge he modifies it, setting the correct label and span. This action implicitly validates a subtree that is composed by the corrected edge plus all its ancestor edges, which we will call the validated prefix tree t_p . When the user replaces the constituent c_{ij}^A with the correct one c_{ij}^B , the validated prefix tree is:

$$t_p(c_{ij}^B) = \{c_{mn}^B : m \leq i, n \geq j, d(c_{mn}^B) \geq d(c_{ij}^A)\} \quad (2)$$

with $d(c_{pq}^D)$ being the depth of constituent c_{pq}^D .

¹In these fields, the user reads the 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.

When a constituent correction is performed, the prefix tree $t_p(c_{ij}^A)$ is fixed and a new tree \hat{t}' that takes into account the prefix is proposed

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

Given that we are working with context-free grammars, the only subtree that effectively needs to be recalculated is the one starting from the parent of the corrected edge. Let the corrected edge be c_{ij}^A and its parent c_{st}^D , then the following tree is proposed

$$\hat{t}' = \arg \max_{t \in \mathcal{T}} p_G(t|\mathbf{x}, t_p) = (\hat{t} \setminus \hat{t}_{st}^D) \cup \hat{t}_{st}^D, \quad (4)$$

with

$$\hat{t}_{st}^D = \arg \max_{t_{st}^D \in \mathcal{T}_{st}} p_G(t_{st}^D|\mathbf{x}_{mn}, c_{ij}^A). \quad (5)$$

Expression (4) represents the newly proposed tree \hat{t}' , which consists of original proposed tree \hat{t} minus the subpart of the original proposed tree \hat{t}_{st}^D (whose root is the parent of the corrected edge c_{st}^D) plus the newly calculated subtree \hat{t}_{st}^D (whose root is also the parent of the corrected constituent c_{st}^D , but also takes into account the corrected one as shown in Expression (5)).

In Figure 1 we show an example that intends to clarify the interactive predictive process. First, the system provides a proposed parse tree (Fig. 1.a). Then the user, which has in his mind the correct reference tree, notices that it has two wrong constituents (c_{23}^X and c_{44}^Z) (Fig. 1.b), and choses to replace c_{23}^X by c_{22}^B (Fig. 1.c). Here, c_{22}^B corresponds to c_{ij}^A from expressions (3) and (5).

As the user does this correction, the system automatically validates the correct prefix: all the ancestors of the modified constituent (dashed line in the figure, $t_p(c_{ij}^A)$ from expression (2)). The system also invalidates the subtrees related to the corrected constituent (dotted line in the figure, \hat{t}_{st}^D from expression (4)).

Finally, the system automatically predicts a new subtree (\hat{t}_{st}^D from expression (4)) (Fig. 1.d). Notice how c_{34}^Z changes its span and c_{44}^D is introduced which provides the correct reference parse.

Within the example shown in Figure 1, the user would obtain the gold tree with just one correction, rather than the three operations needed on a two-step system (one deletion, one substitution and one insertion).

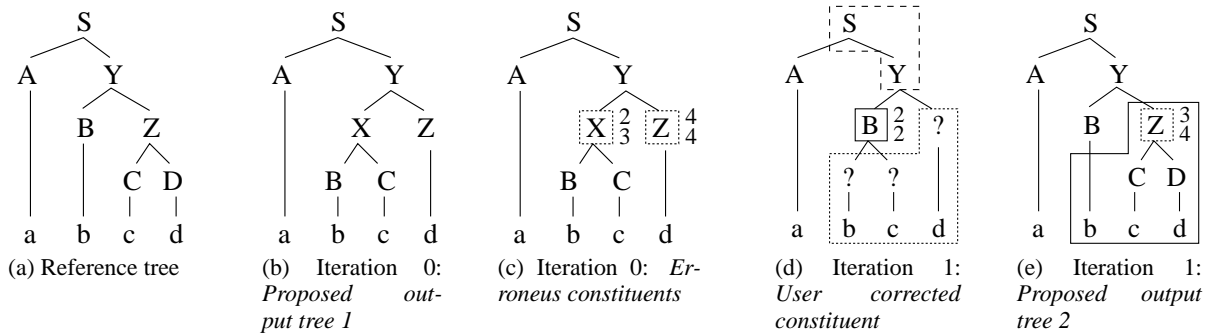


Figure 1: Synthetic example of user interaction with the IPP system.

3 IPP Evaluation

The objective of the experimentation presented here is to evaluate the amount of effort saved for the user using the IPP system, compared to the effort required to manually correct the trees without the use of an interactive system. In this section, we define a standard automatic evaluation protocol, akin to the ones used in Computer-Aided Translation and Computer Aided Text Recognition.

In the absence of testing of an interactive system with real users, the gold reference trees were used to simulate system interaction by a human corrector. In order to do this, the constituents in the proposed tree were automatically reviewed in a preorder manner². In each step, the constituent in the proposed tree was compared to the corresponding one in the reference tree: if the constituent was equivalent no action was taken. When one incorrect constituent was found in the proposed tree, it was replaced by the correct one from the reference tree. This precise step simulated what a human supervisor would do, that is, to type the correct constituent in place of the erroneous one.

The system then performed the predictive step (i.e. recalculation of subtrees related to the corrected constituent). We kept a correction count, which was incremented by one after each predictive step.

3.1 Evaluation metrics

For evaluation, first we report a metric representing the amount of human correcting work needed to obtain the gold tree in a classical two-step process (i.e. the number of operations needed to post-edit the proposed tree in order to obtain the gold

²Interaction in this ordered manner guaranteed that the evaluation protocol only needed to modify the label A and the end point j of a given edge c_{ij}^A , while i remained valid given the modifications of previous constituents.

one). We then compare this value to a metric that measures the amount of effort needed to obtain the gold tree with the human interacting within the presented IPP system.

Parsing quality is generally assessed by the classical evaluation metrics, precision, recall and F-measure. We defined the following metric that measures the amount of effort needed in order to post-edit a proposed tree and obtain the gold reference parse tree, akin to the Word Error Rate used in Statistical Machine Translation and related fields:

- *Tree Constituent Error Rate (TCER)*: Minimum number of constituent substitution, deletion and insertion operations needed to convert the proposed parse tree into the corresponding gold reference tree, divided by the total number of constituents in the reference tree³.

The TCER is in fact strongly related to the F-measure: the higher the F-measure is, the lower TCER will be.

Finally, the relevant evaluation metric that assessed the IPP system performance represents the amount effort that the operator would have to spend using the system in order to obtain the gold tree, and is directly comparable to the TCER:

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

4 Experimental results

An IPP system was implemented over the classical CYK-Viterbi algorithm. Experimentation was run

³Edit distance is calculated over the ordered set of tree constituents. This is an approximation of the edit distance between trees.

over the Penn Tree bank: sections 2 to 21 were used to obtain a vanilla Penn Treebank Grammar; test set was the whole section 23.

We obtained several binarized versions of the train grammar for use with the CYK. The Chomsky Normal Form (CNF) transformation method from the NLTK⁴ was used to obtain several right-factored binary grammars of different sizes⁵.

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

Results for the metrics discussed on section 3.1 for different markovizations of the train grammar can be seen in Table 1. We observe that the percentage of corrections needed using the IPP system is much lower than the rate of needed corrections just post-editing the proposed trees: from 42% to 46% in effort reduction by the human supervisor.

These results clearly show that an interactive predictive system can relieve manual annotators of a lot of burden in their task.

Note that the presented experiments were done using parsing models that perform far from the latest F_1 results; their intention was to assess the utility of the IPP schema. Expected relative reductions with IPP systems incorporating state-of-the-art parsers would not be so large.

PCFG	Baseline		IPP	RelRed
	F_1	TCER	TCAC	
h=0, v=1	0.67	0.40	0.22	45%
h=0, v=2	0.68	0.39	0.21	46%
h=0, v=3	0.70	0.38	0.22	42%

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

5 Conclusions

We have introduced a novel Interactive Predictive Parsing framework which can be operated by a user to obtain error free trees. We have simulated interaction with this system and calculated evaluation metrics, which established that an IPP system results in a high amount of effort reduction for a manual annotator compared to a two-step system.

⁴<http://nltk.sourceforge.net/>

⁵This method implements the vertical (v value) and horizontal (h value) markovizations (Klein and Manning, 2003).

Near term future work includes applying the IPP scenario to state-of-the-art reranking and parsing systems, as well as in the development of adaptive parsing systems

References

- Barrachina, Sergio, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jess Toms, Enrique Vidal, Juan-Miguel Vilar. 2009. *Statistical approaches to computer-assisted translation*. In Computational Linguistics, 35(1) 3-28.
- Charniak, Eugene. 2000. *A maximum-entropy-inspired parser*. In NAACL '00, 132-139.
- Collins, Michael. 2003. *Head-driven statistical models for natural language parsing*. In Computational Linguistics, 29(4):589-637.
- De la Clergerie, Éric, Olivier Hamon, Djamel Mostefa, Christelle Ayache, Patrick Paroubek and Anne Vilnat. 2008. *PASSAGE: from French Parser Evaluation to Large Sized Treebank*. In LREC'08.
- Huang, Liang. 2008. *Forest reranking: discriminative parsing with non-local features*. In ACL '08.
- Johnson, Mark. 1998. *PCFG models of linguistic tree representation*. In Computational Linguistics, 24:613-632.
- Klein, Dan and Christopher D. Manning. 2003. *Accurate Unlexicalized Parsing*. In ACL '03, 423-430.
- Lease, Matthew, Eugene Charniak, Mark Johnson and David McClosky. 2006. *A look at parsing and its applications*. In National Conference on Artificial Intelligence, vol. 21-II, 1642-1645.
- Marcus, Mitchell P., Mary Ann Marcinkiewicz and Beatrice Santorini. 1995. *Building a Large Annotated Corpus of English: The Penn Treebank*. Computational Linguistics 19(2), 313-330.
- Matsuzaki, Takuya, Yasuke Miyao and Jun'ichi Tsujii. 2005. *Probabilistic CFG with latent annotations*. In ACL '05, 75-82.
- McClosky, David, Eugene Charniak and Mark Johnson. 2006. *Effective self-training for parsing*. In HLT-NAACL '06
- Petrov, Slav and Dan Klein. 2007. *Improved inference for unlexicalized parsing*. In NAACL-HLT '07.
- Toselli, Alejandro, Verónica Romero and Enrique Vidal. 2008. *Computer Assisted Transcription of Text Images and Multimodal Interaction*. In MLMI '08.
- Vidal, Enrique, Francisco Casacuberta, Luis Rodriguez, Jorge Civera and Carlos D. Martinez Hinarejos. 2006. *Computer-assisted translation using speech recognition*. In IEEE Trans. on Audio, Speech, and Language Processing, 14(3), 941-951.
- Yamamoto, Ryo, Shinji Sako, Takuya Nishimoto and Shigeki Sagayama. 2006. *On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar*. In 10th International Workshop on Frontiers in Handwriting Recognition.