# Statistical Confidence Measures for Probabilistic Parsing

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

We introduce a formal framework that allows the calculation of new purely statistical confidence measures for parsing, which are estimated from posterior probability of constituents. These measures allow us to mark each constituent of a parse tree as correct or incorrect. Experimental assessment using the Penn Treebank shows favorable results for the classical confidence evaluation metrics: the CER and the ROC curve. We also present preliminar experiments on application of confidence measures to improve parse trees by automatic constituent relabeling.

## 1 Introduction

Many parsing methods exist in the literature, including those based on Probabilistic Context-Free Grammars (PCFGs). Great effort has been undertaken to improve performance of these parsers. First, lexicalization of grammars with elaborate smoothing accomplished very promising results [4, 5]. Then, manual tree annotation and nonterminal spliting greatly shortened the gap between unlexicalized models and their better performing lexicalized counterparts [10, 12]. Later, automatic tree annotation systems, using a nonterminal split-and-merge approach and a hierarchy of progressively refined grammars, provided superior results over the best lexicalized approaches [14, 16, 17]. Last but not least, the most impressive results were achieved by reranking systems, as shown in the semi-supervised method of [15], or the forest reranking approach of [9] which uses packed parse forests (compact structures that contain many possible tree derivations).

Given the difficulty and importance of parsing in all of its applications [13], there exists an increasing necessity to detect erroneous syntactic structures therein. This need is even more present in parse trees that are obtained using current high performing systems, especially if error-free trees are desired. In such a case, the few remaining erroneous parts need to be quickly detected and manually corrected (possibly using interactive methods). Assessing the correctness of the different parts of the parsing is needed for the construction of efficient computer-assisted interactive predictive parsing systems, which will be useful in the creation of new gold standard treebanks [6]. This paper is a step forward in introducing Confidence Measures into the parsing world.

Confidence measures are a powerful formalism that have been used to detect individual erroneous words in Automatic Speech Recognition (ASR) and Statistical Machine Translation (SMT) output sentences [19, 18]. Once these errors are detected and marked, they can be more easily corrected, either by automatic or manual methods.

In parsing, confidence measures detect erroneous constituents. Some confidence measures for parsing in the form of combinations of characteristics calculated from n-best lists were proposed in [2]. In our work, we present an alternative more akin to word graph-based methods in ASR and SMT.

Other works have proposed to improve parsing results by defining parsing algorithms that try to maximize alternative objective functions. In [8], Goodman derived an algorithm that maximized the labeled recall evaluation criterion (rather than maximizing the whole tree probability as the classical CYK-Viterbi does) which presents some similarities with the confidence measure framework presented here.

Goodman's algorithm presented the problem of producing trees that were not grammatical, and as such, unsuitable for downstream processing. However, many applications can benefit from maximizing the number of correct constituents, regardless of the grammaticality of the tree, for example, machine translation systems.

The *max-rule* parser, which is a variation of Goodman's algorithm that solves the ungrammaticality issue, has been used in very recent top performing parsing systems [14, 17]. In [17], the authors also proposed different objective functions for parsing with posterior probabilities.

The performance of our proposal is assessed by classical metrics, the Confidence Error Rate (CER) and the Receiver Operating Characteristic (ROC) curve, which are widely used for confidence measure evaluation [19, 18]. Additionally, we introduce experimentation exemplifying the use of confidence measures for automatic constituent relabeling for the improvement of $F_1$ and POS tag accuracy results.

## 2 Statistical confidence measures

In ASR and SMT, confidence measures refer to the probability of single words being correct in an output sentence, and they are mostly calculated from the posterior probability of each word.

One way to estimate the posterior probability is to use n-best lists. In this case, the probability of a word being correct is determined by how many times the word appears in a similar position over all the n-best sentences.

More recently, the posterior probability is obtained using a forward-backward expression over word graphs [19, 18]. Word graphs can be seen as a condensing of the information contained in an n-best list. In the ideal case of a non-pruned word graph, it represents all the possible out-



*International Conference RANLP 2009 - Borovets, Bulgaria*, pages 388–392

put sentences for a given input. In practice, word graphs are usually pruned, so they contain only information about the most probable outputs. This approach presents greater flexibility than n-best lists since word graphs are not limited by a predefined number of $n$ outputs, but rather take form depending on the concentration of probability mass.

### 2.1 Edge posterior probability

A tree $t$ is composed of substructures that are usually referred to as constituents or edges. Given a tree $t$ associated to a string $x_{1|x|}$, a constituent $c_{ij}^A$ is defined by a nonterminal symbol (or syntactic tag) $A$ that spans the substring $x_{ij}$.

In this paper, we establish a framework for probabilistic calculation of confidence measures for edges $c_{ij}^A$, which uses edge posterior probability. This is similar to the calculation of posterior probability over word graphs and its use as a confidence measure presented in SMT [18].

Let $G$ be a probabilistic Context-Free Grammar, and let $\boldsymbol{x} = x_1 \dots x_{|x|}$ an input sentence. The parser analyzes the input sentence $\boldsymbol{x} = x_1 \dots x_{|x|}$ and then produces the most probable parse tree $\hat{t} = \arg\max_{t \in \mathcal{T}} p_G(t|\boldsymbol{x})$, where $p_G(t|\boldsymbol{x})$ is the probability of the tree, and $\mathcal{T}$ is the set of all possible parse trees for $\boldsymbol{x}$.

The posterior probability of a constituent can be considered as a measure of the degree to which the constituent is believed to be correct. The posterior probability of a constituent given the string $\boldsymbol{x}$ is

$$p_G(c_{ij}^A|\boldsymbol{x}) = \frac{p_G(c_{ij}^A, \boldsymbol{x})}{p_G(\boldsymbol{x})} = \frac{\sum_{t' \in \mathcal{T}:\, c_{ij}^A \in t'} p_G(t'|\boldsymbol{x})}{p_G(\boldsymbol{x})}\,, \qquad (1)$$

that is, the normalized probability of the constituent $c_{ij}^A$ being placed on the tree in the exact position that spans the $x_i \dots x_j$ substring. The upper part is the sum of probabilities of all possible parse trees for $\boldsymbol{x}$ containing the nonterminal $A$ with the same exact start and end points $i$ and $j$.

Eq. (1) can be efficiently computed with the inside ($\beta_A(i,\, j) = p_G(A \Rightarrow^* x_i \dots x_j)$) and outside ($\alpha_A(i,\, j) = p_G(S \Rightarrow^* x_1 \dots x_{i-1}\, A\, x_{j+1} \dots x_{|\boldsymbol{x}|})$) probabilities introduced in [1] (see Fig. 1):

$$p_G(c_{ij}^A|\boldsymbol{x}) = \frac{\beta_A(i,\, j)\; \alpha_A(i,\, j)}{\beta_S(1,\, |\boldsymbol{x}|)}\,. \qquad (2)$$

The posterior probability can now directly be used as a measure of the confidence in each individual edge

$$\mathcal{C}(c_{ij}^A) = p_G(c_{ij}^A|\boldsymbol{x})\,. \qquad (3)$$

Eq. (2) is the same expression that is maximized in [8] for the labeled recall parsing algorithm, which can indeed be seen as a confidence measure-based parsing algorithm.

Fig. 2 shows a synthetic example in order to clarify the confidence measure concept. This figure shows the only four possible parse trees for the string $abc$. Let all productions in the grammar of the example carry the same probability, and suppose that the parser returns *(a)* tree. Then the following confidence measure values are obtained for the edges in the *(a)* tree: $\mathcal{C}(c_{13}^S) = 1$, $\mathcal{C}(c_{12}^Z) = 2/4$, $\mathcal{C}(c_{11}^A) = 1$, $\mathcal{C}(c_{22}^B) = 1$ and $\mathcal{C}(c_{33}^D) = 1/4$. If the correct parse tree is *(e)*, which is unobtainable by the example grammar, then setting a confidence threshold would allow us to know that the $c_{33}^D$ edge is incorrect in the *(a)* tree.

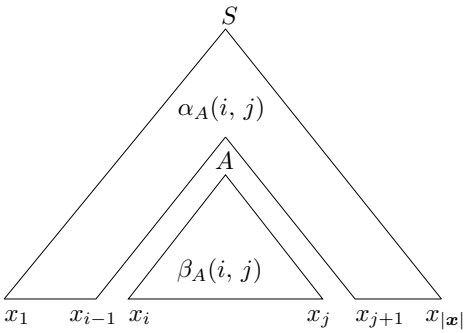


Fig. 1: The product of the the inside probability $\beta_A(i,\, j) = p_G(A \Rightarrow^* x_i \dots x_j)$ and the outside probability $\alpha_A(i,\, j) = p_G(S \Rightarrow^* x_1 \dots x_{i-1}\, A\, x_{j+1} \dots x_{|\boldsymbol{x}|})$, comprises the upper part of expression (2)

## 3 Experiments

In the experiments presented in this section, we show how confidence measures can help parsing through the detection of erroneous constituents. We introduce evaluation metrics that assess the performance of confidence measures in section 3.1, we define the experimental framework on section 3.2, and we present empirical results on section 3.3. Additionally, we introduce experimentation showing how confidence measures can be used for tree improvement by automatic constituent relabeling in section 3.4.

### 3.1 Evaluation metrics

Performance of a confidence measure refers to its ability to detect erroneous constituents. We report results on two classical metrics: the Confidence Error Rate (CER) and the Receiver Operating Characteristic (ROC) curve with its corresponding integrated area (IROC) [19, 18]. The results for these metrics are presented for both syntactic constituents and POS tag together as well as separately. At this point, the F-measure cannot be used to evaluate the performance of confidence scores because there is only one set of parse trees with confidence scores attached. Two sets of parse trees are necessary for F-measure comparison, so it is not reported until section 3.4.

Given a tree with a number of constituents $n$ (some correct and some incorrect) and a confidence score attached to each one, each constituent is marked as either correct or incorrect depending on whether its confidence exceeds the confidence threshold $\tau$, which is obtained beforehand using a development set.

The $CER(\tau) = \frac{n_{fr}(\tau)+n_{fa}(\tau)}{n}$ is the total number of incorrect marks divided by the total number of constituents (false rejection $n_{fr}(\tau)$ is the number of constituents that are correctly obtained by the parser but that are deemed incorrect by the confidence measure; false acceptance $n_{fa}(\tau)$ is the number of erroneus constituents marked correct due to their high confidence value).

In the ideal case of perfect confidence measures, incorrect and correct constituents are discriminated without mistakes and the CER is zero. The baseline CER is the one obtained assuming that all syntactic edges are correct (the only possible assumption when confidence measures are not available), it is the number of erroneous constituents

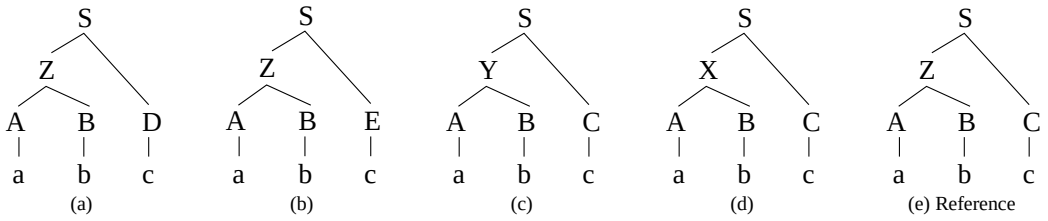


Fig. 2: Synthetic example of a confidence measure calculation. Assume that all productions in the grammar have the same probability. The grammar can only generate the *(a)*, *(b)*, *(c)* and *(d)* parse trees for the $abc$ input string. The reference parse tree is unobtainable. Confidence measures for the edges in the $(a)$ tree are $\mathcal{C}(c_{13}^{S}) = 1,\ \mathcal{C}(c_{12}^{Z}) = 2/4,\ \mathcal{C}(c_{11}^{A}) = 1,\ \mathcal{C}(c_{22}^{B}) = 1$ and $\mathcal{C}(c_{33}^{D}) = 1/4$

divided by the total.

Another measure that determines the goodness of confidence measures globally over all possible thresholds is the ROC curve, which is the plot of the correct rejection rate against the correct acceptance rate for all possible values of $\tau \in [0, 1]$. The worst case ROC is a diagonal line, and the further it lies from the diagonal towards $1.0$ on both axes, the better the ROC is. A ROC curve provides a qualitative analysis of the adequacy of the confidence measure. Its corresponding IROC (integrated area under a ROC curve taking values in the interval $[0, 1]$) accounts for the corresponding quantitative metric.

Once incorrect constituents are detected, actions to correct them can be carried out. In Section 3.4, we present some experimentation that does this by automatic relabeling incorrect constituents.

## 3.2 Experimental framework

Standard train and test splits were defined over the Penn Tree bank. Sections 2 to 21 were used to obtain a vanilla Penn Treebank Grammar; the test set was the whole section 23; and the development set was comprised of the first 346 sentences of section 24.

Since CYK works with grammars in the Chomsky Normal Form (CNF), we obtained several binarized versions of the train grammar. We used the CNF transformation method from the open source NLTK[1] to obtain several right-factored binary grammars of different sizes. This method implements the vertical ($v$ value) and horizontal ($h$ value) markovizations [12].

We modified the CYK to perform the confidence measure calculation at parsing time, using equation (1) as described in section 2.

For out-of-vocabulary words, when an input word could not be derived by any of the preterminals in the treebank grammar, a very small probability for that word was uniformly added to all of the preterminals.

An unbinarization process was performed over the obtained parse trees in order to compare them to the reference trees. Newly introduced nonterminals were removed, and their children became attached to their original parents.

The constituents in each proposed solution tree were then automatically compared to the ones in the gold-standard corpus. Each constituent was marked as correct or incorrect depending on whether or not the corresponding constituent existed in the reference tree.

[1] http://nltk.sourceforge.net/

With the edges labeled as either correct or incorrect, the baseline CER and the confidence measure CER were calculated for the test set. Since the CER depends on the selected threshold, the separate development set was used to obtain the best threshold. ROC curves with their IROC values for the test set were also calculated.

## 3.3 CER and ROC results

We calculated metric results for the presented confidence measure using the three different markovizations of the train grammar shown in Table 1. Increasing the $v$ markovization parameter produces better performing PCFGs, but also increases the number of nonterminals. When parsed with these grammars, the 346 sentences in the development set produced about 16k elements (7k syntactic constituents and 9k POS tags), and the 2416 sentences in the test set produced about 101k ones (44k syntactic constituents and 57k POS tags).

| PCFG | Size |
|---|---|
| h=0,v=1 | 561 |
| h=0,v=2 | 2,034 |
| h=0,v=3 | 5,058 |

Table 1: Grammar size after each markovization (number of nonterminals).

Performance of the confidence measure is reflected in the classical confidence evaluation metrics discussed in section 3.1: improvement of the best CER over the baseline CER; the ROC curve, and its corresponding IROC. The results for the test set are presented in Table 2.

The confidence measures are able to discriminate an high number of incorrect constituents, as show by the clear improvements over the baseline CER for all markovizations of the PCFG, both for syntactic constituents and for POS tags.

Even for the PCFG with the best baseline CER (*h=0,v=3*), the confidence measures allowed us to detect that 2.4% of the edges could be erroneously labeled (4.9% of syntactic constituents, and 1% of POS tags), this is a relative reduction of 14.6% (15.9% for syntactic constituents, and 20% for POS tags). The ROC curves for the mentioned PCFG are presented in Fig. 3. This figure shows that the confidence measures discriminate better over POS tags than over syntactic constituents, which is consistent with the baselines and relative CER gains for each category of constituents.

| | $F_1$ | | | POS tag accuracy | | |
|---|---|---|---|---|---|---|
| PCFG | Basel. | Relabel. | $\Delta$ | Basel. | Relabel. | $\Delta$ |
| h=0,v=1 | 67.87 | 68.01 | .14±.08 | 96.11 | 96.35 | .24±.11 |
| h=0,v=2 | 71.09 | 71.20 | .11±.07 | 96.30 | 96.54 | .24±.09 |
| h=0,v=3 | 71.17 | 71.31 | .14±.07 | 96.23 | 96.51 | .28±.10 |

Table 3: $F_1$ and POS tag accuracy for the test set: baseline scores, relabeling scores, and increments. Accuracy values are bootstrap estimates with $B = 10^4$; the improvement interval is a 95% confidence interval based on the standard error estimate [3].

| | | Basel. | Confidence M. | | |
|---|---|---|---|---|---|
| PCFG | TAGS | CER | CER | RelR | IROC |
| h=0 | all | 17.8 | 12.3 | 30.9% | 0.81 |
| v=1 | syn | 34.3 | 22.8 | 33.5% | 0.65 |
| | pos | 5.2 | 4.2 | 19.2% | 0.86 |
| h=0 | all | 16.4 | 13.2 | 19.5% | 0.77 |
| v=2 | syn | 31.1 | 24.6 | 20.9% | 0.57 |
| | pos | 5.0 | 4.4 | 12.0% | 0.86 |
| h=0 | all | 16.4 | 14.0 | 14.6% | 0.75 |
| v=3 | syn | 30.9 | 26.0 | 15.9% | 0.50 |
| | pos | 4.9 | 4.5 | 8.1% | 0.86 |

Table 2: Metric results for the test set: baseline CER, confidence CER (with the best development threshold), CER relative reduction, and IROC for each PCFG.

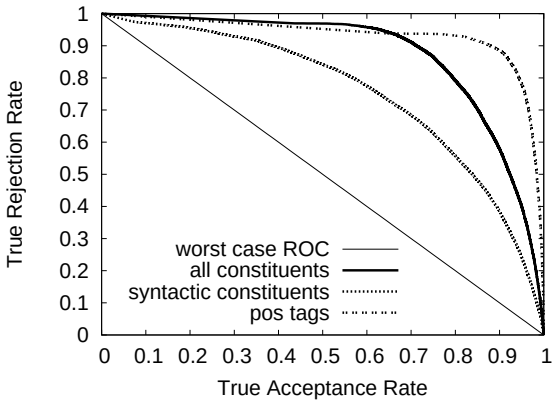


Fig. 3: ROC curves for markovization *h=0,v=3*.

Our results can be compared to the ones presented in [2], in which confidence measures were calculated from n-best lists obtained by the Charniak parser. Comparing the CERs presented here to the ones shown in the cited work, we observe that our relative reductions are consistently higher. Note that, in our work, we carried out unlexicalized parsing; therefore, our baseline CERs are slightly worse than the ones reported in the cited paper.

### 3.4 Confidence measures for automatic constituent relabeling

Finally, we employed confidence measures in an experiment that consisted of improving trees by constituent relabeling.

After obtaining the best parse tree, the confidence value of each available nonterminal was calculated for each element (both syntactic constituent and POS tags) position and span. The nonterminal that yielded the maximum confidence value was introduced as the new label of the constituent. As we mentioned above, this process does not guarantee the grammaticality of the resulting trees.

The results are shown in Table 3. We obtained small but statistically significant (95% confidence intervals as in [3]) improvements, not only in POS tag accuracy but also in LP/LR $F_1$.

In syntactic tags, the advantage obtained by our system is marginal. This is possibly due to the more severe structural errors present in the start and end points of the bracketings. A better approach would be to completely discard groups of incorrect constituents and calculate completely new ones.

Although the results presented in this section are far from the current state of the art, the improvements presented here both exemplify one of the possible uses of confidence measures and support the good confidence measure metric results presented in section 3.3. These experiments discover a new path that is worth exploring in order to achieve further parsing improvements.

## 4 Conclusions

A new formal framework for calculating a purely statistical confidence measure (based on inside-outside estimated posterior probability of constituents) for probabilistic parsing has been introduced.

Experiments were performed on the Penn Treebank: CERs showed that the proposed confidence measure is able to discriminate a high number of correct constituents from incorrect ones. This is confirmed by similarly good IROC values. The relabeling experiment also resulted in consistent improvements in $F_1$ and POS tag accuracy.

Future work involves using confidence measures for improving state-of-the-art parsing and reranking systems as well as building efficient computer-aided predictive interactive parsing systems.

## Acknowledgements

Work supported by the EC (FEDER) and the Spanish MEC under the MIPRCV "Consolider Ingenio 2010" research programme (CSD2007-00018), the iTransDoc research project (TIN2006-15694-CO2-01), and the FPU fellowship AP2006-01363.

## References


[1] Baker, J. 1979. *Trainable grammars for speech recognition*. In Speech Communications, MASA'79, 31-35.

[2] Benedí, José-Miguel, Joan-Andreu Sánchez and Alberto Sanchís. 2007. *Confidence measures for stochastic parsing*. In RANLP '07, 58-63,.

[3] Bisani, Maximilian and Hermann Ney. 2004. *Bootstrap estimates for confidence intervals in asr performance evaluation*. In ICASSP '04, I:409-412.

[4] Charniak, Eugene. 2000. *A maximum-entropy-inspired parser*. In NAACL '00, 132-139.

[5] Collins, Michael. 2003. *Head-driven statistical models for natural language parsing*. In Computational Linguistics, 29(4):589-637.

[6] 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.

[7] Earley, Jay. 1970. *An efficient context-free parsing algorithm*. In Communications of the ACM'70, 8(6):451-455.

[8] Goodman, Joshua. 1996. *Parsing algorithms and metrics*. In ACL '96, 177-183.

[9] Huang, Liang. 2008. *Forest reranking: discriminative parsing with non-local features*. In ACL '08.

[10] Johnson, Mark. 1998. *PCFG models of linguistic tree representation*. In Computational Linguistics '98, 24:613-632.

[11] Klein, Dan and Chistopher D. Manning. 2001. *Parsing with treebank grammars: Empirical bounds, theoretical models, and the structure of the Penn treebank*. In ACL '01, 338-345.

[12] Klein, Dan and Chistopher D. Manning. 2003. *Accurate Unlexicalized Parsing*. In ACL '03, 423-430.

[13] Lease, Matthew, Eugene Charniak, Mark Johnson and David McClosky. 2006. *A look at parsing and its applications*. In National Conference on Artificial Intelligence '96, vol. 21-II, 1642-1645.

[14] Matsuzaki, Takuya, Yasuke Miyao and Jun'ichi Tsujii. 2005. *Probabilistic CFG with latent annotations*. In ACL '05, 75-82.

[15] McClosky, David, Eugene Charniak and Mark Johnson. 2006. *Effective self-training for parsing*. In HLT-NAACL '06.

[16] Petrov, Slav, Leon Barrett, Romain Thibaux and Dan Klein. 2006. *Learning accurate, compact and interpretable tree annotation*. In ACL '06, 433-440.

[17] Petrov, Slav and Dan Klein. 2007. *Improved inference for unlexicalized parsing*. In NAACL-HLT '07.

[18] Ueffing, Nicola and Hermann Ney. 2007. *Word-level confidence estimation for machine translation*. In Computational Linguistics 33(1), 9-40.

[19] Wessel, Frank, Ralf Schlüter, Klaus Macherey and Hermann Ney. 2001. *Confidence measures for large vocabulary continuous speech recognition*. In IEEE Tran. on Speech and Audio Processing, 3(9):288-298.