Computing Confidence Scores for All Sub Parse Trees

Feng Lin

Department of Computer Science and Engineering
Fudan University
Shanghai 200433, P.R. China
fenglin@fudan.edu.cn

Fuliang Weng

Research and Technology Center
Robert Bosch LLC
Palo Alto, CA, 94303, USA
fuliang.weng@us.bosch.com

Abstract

Computing confidence scores for applications, such as dialogue system, information retrieving and extraction, is an active research area. However, its focus has been primarily on computing word-, concept-, or utterance-level confidences. Motivated by the need from sophisticated dialogue systems for more effective dialogs, we generalize the confidence annotation to all the subtrees, the first effort in this line of research. The other contribution of this work is that we incorporated novel long distance features to address challenges in computing multi-level confidence scores. Using Conditional Maximum Entropy (CME) classifier with all the selected features, we reached an annotation error rate of 26.0% in the SWBD corpus, compared with a subtree error rate of 41.91%, a closely related benchmark with the Charniak parser from (Kahn et al., 2005).

1 Introduction

There has been a good amount of interest in obtaining confidence scores for improving word or utterance accuracy, dialogue systems, information retrieving & extraction, and machine translation (Zhang and Rudnicky, 2001; Guillevic et al., 2002; Gabsdil et al., 2003; Ueffing et al., 2007).

However, these confidence scores are limited to relatively simple systems, such as command-n-control dialogue systems. For more sophisticated dialogue systems (e.g., Weng et al., 2007), identification of reliable phrases must be performed at different granularity to ensure effective and friendly dialogues. For example, in a request of MP3 music domain "Play a rock song by Cher", if we want to communicate to the user that the system is not confident of the phrase "a rock song," the confidence scores for each word, the artist name "Cher," and the whole sentence would not be enough. For tasks of information extraction, when extracted content has internal structures, confidence scores for such phrases are very useful for reliable returns.

As a first attempt in this research, we generalize confidence annotation algorithms to all sub parse trees and tested on a human-human conversational corpus, the SWBD. Technically, we also introduce a set of long distance features to address the challenges in computing multi-level confidence scores.

This paper is organized as follows: Section 2 introduces the tasks and the representation for parse trees; Section 3 presents the features used in the algorithm; Section 4 describes the experiments in the SWBD corpus; Section 5 concludes the paper.

2 Computing Confidence Scores for Parse Trees

The confidence of a sub-tree is defined as the posterior probability of its correctness, given all the available information. It is $P(sp\,is\,correct|\,x)$ – the posterior probability that the parse sub-tree *sp* is correct, given related information *x*. In real applications, typically a threshold or cutoff *t* is needed:

$$sp\ is \begin{cases} correct,\ if\ P(sp\,is\,correct|\,x) \geq t \\ incorrec,\ if\ \ P(sp\,is\,correct|\,x) < t \end{cases} \quad (1)$$

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 217–220,
Columbus, Ohio, USA, June 2008.

In this work, the probability $P(sp\,is\,correct|\,x)$ is calculated using CME modeling framework:

$$P(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_j \lambda_j f_j(x, y) \right) \quad (2)$$

where $y \in \{sp\ is\ correct,\ sp\ is\ incorrect\}$, x is the syntactic context of the parse sub-tree sp, f_j are the features, λ_j are the corresponding weights, and $Z(x)$ is the normalization factor.

The parse trees used in our system are lexicalized binary trees. However, the confidence computation is independent of any parsing method used in generating the parse tree as long as it generates the binary dependency relations. An example of the lexicalized binary trees is given in Figure 1, where three important components are illustrated: the left sub-tree, the right sub-trees, and the marked head and dependency relation.

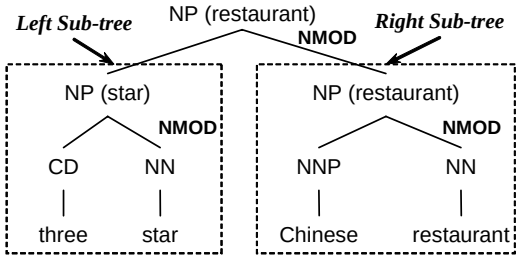

Figure 1. Example of parse sub-tree's structure for phrase "three star Chinese restaurant"

Because the parse tree is already given, a bottom-up left-right algorithm is used to traverse through the parse tree: for each subtree, compute its confidence, and annotate it as correct or wrong.

3 Features

Four major categories of features are used, including, words, POS tags, scores and syntactic information. Due to the space limitation, we only give a detailed description of the most important one[1], lexical-syntactic features.

The lexical-syntactic features include lexical, POS tag, and syntactic features. Word and POS tag features include the head and modifier words of the parse sub-tree and the two children of the root, as well as their combinations. The POS tags and hierarchical POS tags of the corresponding words are also considered to avoid data sparseness. The adopted hierarchical tags are: Verb-related (V), Noun-related (N), Adjectives (ADJ), and Adverbs (ADV), similar to (Zhang et al, 2006).

Long distance structural features in statistical parsing lead to significant improvements (Collins et al., 2000; Charniak et al., 2005). We incorporate some of the reported features in the feature space to be explored, and they are enriched with different POS categories and grammatical types. Two examples are given below.

One example is the Single-Level Joint Head and Dependency Relation (SL-JHD). This feature is pairing the head word of a given sub-tree with its last dependency relation. To address the data sparseness problem, two additional SL-JHD features are considered: a pair of the POS tag of the head of a given sub-tree and its dependency relation, a pair of the hierarchical POS tag of the head of a given sub-tree and its dependency relation. For example, for the top node in Figure 2, (restaurant NCOMP), (NN, NCOMP), and (N, NCOMP) are the examples for the three SL-JHD features. To compute the confidence score of the sub-tree, we include the three JHD features for the top node, and the JHD features for its two children. Thus, for the sub-tree in Figure 2, the following nine JHD features are included in the feature space, i.e., (restaurant NCOMP), (NN, NCOMP), (N, NCOMP), (restaurant NMOD), (NN NMOD), (N NMOD), (with POBJ), (IN POBJ), and (ADV POBJ).

The other example feature is Multi-Level Joint Head and Dependency Relation (ML-JHD), which takes into consideration the dependency relations at multiple levels. This feature is an extension of SL-JHD. Instead of including only single level head and dependency relations, the ML-JHD feature includes the hierarchical POS tag of the head and dependency relations for all the levels of a given sub-tree. For example, given the sub-tree in Figure 3, (NCOMP, N, NMOD, N, NMOD, N, POBJ, ADV, NMOD, N) is the ML-JHD feature for the top node (marked by the dashed circle).

In addition, three types of features are included: dependency relations, neighbors of the head of the current subtree, and the sizes of the sub-tree and its left and right children. The dependency relations include the top one in the subtree. The neighbors are typically within a preset distance from the head word. The sizes refer to the numbers of words or non-terminals in the subtree and its children.

[1] The other important one is the dependency score, which is the conditional probability of the last dependency relation in the subtree, given its left and right child trees

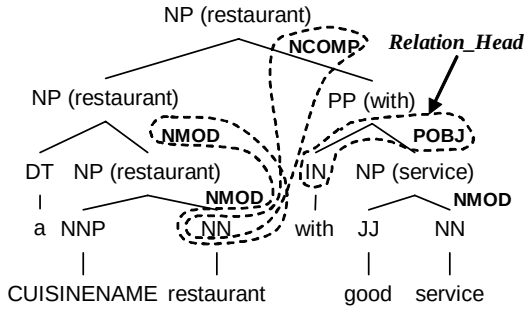

Figure 2. SL-JHD Features

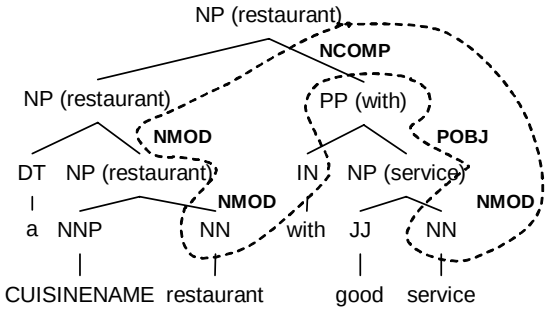

Figure 3. ML-JHD Features

4 Experiments

Experiments were conducted to see the performance of our algorithm in human to human dialogs – the ultimate goal of a dialogue system. In our work, we use a version of the Charniak's parser from (Aug. 16, 2005) to parse the re-segmented SWBD corpus (Kahn et al., 2005), and extract the parse sub-trees from the parse trees as experimental data.

The parser's training procedure is the same as (Kahn et al., 2005). The only difference is that they use golden edits in the parsing experiments while we delete all the edits in the UW Switchboard corpus. The F-score of the parsing result of the Charniak parser without edits is 88.24%.

The Charniak parser without edits is used to parse the training data, testing data and tuning data. We remove the sentences with only one word and delete the interjections in the hypothesis parse trees. Finally, we extract parse sub-trees from these hypothesis parse trees. Based on the gold parse trees, a parse sub-tree is labeled with 1 (correct), if it has all the words, their POS tags and syntactic structures correct. Otherwise, it is 0 (incorrect). Among the 424,614 parse sub-trees from the training data, 316,182 sub-trees are labeled with 1; among the 38,774 parse sub-trees from testing data, 22,521 ones are labeled with 1; and among the 67,464 parse sub-trees from the tuning data, 38,619 ones are labeled with 1. In the testing data, there are 5,590 sentences, and the percentage of complete bracket match[2] is 57.11%, and the percentage of parse sub-trees with correct labels at the sentence level is 48.57%. The percentage of correct parse sub-trees is lower than that of the complete bracket match due to its stricter requirements.

Table 1 shows our analysis of the testing data. There, the first column indicates the phrase length categories from the parse sub-trees. Among all the parse trees in the test data, 82.84% (first two rows) have a length equal to or shorter than 10 words. We converted the original parse sub-trees from the Charniak parser into binary trees.

Length	Sub-tree Types	Number	Ratio
<=10	Correct	21,593	55.70%
	Incorrect	10,525	27.14%
>10	Correct	928	2.39%
	Incorrect	5,728	14.77%

Table 1. The analysis of testing data.

We apply the model (2) from section 2 on the above data for all the following experiments. The performance is measured based on the confidence annotation error rate (Zhang and Rudnicky, 2001).

$$Annot.Error = \frac{Number Of\ Subtrees Annotated As Incorrect}{Total\ Number\ Of\ Subtrees}$$

Two sets of experiments are designed to demonstrate the improvements of our confidence computing algorithm, as well as the newly introduced features (see Table 2 and Table 3).

Experiments were conducted to evaluate the effectiveness of each feature category for the sub-tree level confidence annotation on SWBD corpus (Table 2). The baseline system uses the conventional features: words and POS tags. Additional feature categories are included separately. The syntactic feature category shows the biggest improvement among all the categories.

To see the additive effect of the feature spaces for the multi-level confidence annotation, another set of experiments were performed (Table 3). Three feature spaces are included incrementally: dependency score, hierarchical tags and syntactic features. Each category provides sizable reduction in error rate. Totally, it reduces the error rate by

[2] Complete bracket match is the percentage of sentences where bracketing recall and precision are both 100%.

	Feature Space Description	Annot. Error	Relative Error Decrease
Baseline	Base features: Words, POS tag	36.2%	\
Set 1	Base features + Dependency score	32.8%	9.4%
Set 2	Base features + Hierarchical tags	35.3%	2.5%
Set 3	Base features + Syntactic features	29.3%	19.1%

Table 2. Comparison of different feature space (on SWBD corpus).

	Feature Space Description	Annot. Error	Relative Error Decrease
Baseline	Base features: Words, POS tag	36.2%	\
Set 4	+ Dependency score	32.8%	9.4%
Set 5	+ Dependency score + hierarchical tags	32.7%	9.7%
Set 6	+ Dependency score + hierarchical tags + syntactic features	26.0%	28.2%

Table 3. Summary of experiment results with different feature space (on SWBD corpus).

10.2%, corresponding to 28.2% of a relative error reduction over the baseline. The best result of annotation error rate is 26% for Switchboard data, which is significantly lower than the 41.91% sub-tree parsing error rate (see Table 1: 41.91% = 27.14%+14.77%). So, our algorithm would also help the best parsing algorithms during rescoring (Charniak et al., 2005; McClosky et al., 2006).

We list the performance of the parse sub-trees with different lengths for Set 6 in Table 4, using the F-score as the evaluation measure.

Length	Sub-tree Category	F-score
<=10	Correct	82.3%
	Incorrect	45.9%
>10	Correct	33.1%
	Incorrect	86.1%

Table 4. F-scores for various lengths in Set 15.

The F-score difference between the ones with correct labels and the ones with incorrect labels are significant. We suspect that it is caused by the different amount of training data. Therefore, we simply duplicated the training data for the sub-trees with incorrect labels. For the sub-trees of length equal to or less than 10 words, this training method leads to a 79.8% F-score for correct labels, and a 61.4% F-score for incorrect labels, which is much more balanced than those in the first set of results.

5 Conclusion

In this paper, we generalized confidence annotation algorithms to multiple-level parse trees and demonstrated the significant benefits of using long distance features in SWBD corpora. It is foreseeable that multi-level confidence annotation can be used for many other language applications such as parsing, or information retrieval.

References

Eugene Charniak and Mark Johnson. 2005. *Coarse-to-fine n-best parsing and MaxEnt discriminative reranking*. Proc. ACL, pages 173–180.

Michael Collins. 2000. *Discriminative reranking for natural language parsing*. Proc. ICML, pages 175–182.

Malte Gabsdil and Johan Bos. 2003. *Combining Acoustic Confidence Scores with Deep Semantic Analysis for Clarification Dialogues*. Proc. IWCS, pages 137-150.

Didier Guillevic, et al. 2002. *Robust semantic confidence scoring*. Proc. ICSLP, pages 853-856.

Jeremy G. Kahn, et al. 2005. *Effective Use of Prosody in Parsing Conversational Speech.* Proc. EMNLP, pages 233-240.

David McClosky, Eugene Charniak and Mark Johnson. 2006. *Reranking and Self-Training for Parser Adaptation.* Proc. COLING-ACL, pages 337-344.

Nicola Ueffing and Hermann Ney. 2007. *Word-Level Confidence Estimation for Machine Translation.* Computational Linguistics, 33(1):9-40.

Fuliang Weng, et al., 2007. *CHAT to Your Destination.* Proc. of the 8th SIGDial workshop on Discourse and Dialogue, pages 79-86.

Qi Zhang, Fuliang Weng and Zhe Feng. 2006. *A Pro-gressive Feature Selection Algorithm for Ultra Large Feature Spaces.* Proc. COLING-ACL, pages 561-568.

Rong Zhang and Alexander I. Rudnicky. 2001. *Word level confidence annotation using combinations of features.* Proc. Eurospeech, pages 2105-2108.