# ROBUST DATA ORIENTED PARSING OF SPEECH UTTERANCES

**Khalil Sima'an**
ILK, Tilburg University + ILLC, University of Amsterdam
Spuistraat 134, 1012 VB, Amsterdam, The Netherlands
khalil.simaan@hum.uva.nl

**Abstract**

Spoken utterances do not always abide by linguistically motivated grammatical rules. These utterances exhibit various phenomena considered outside the realm of theoretically-oriented linguistic research. For a language model that extends linguistically motivated grammars with probabilistic reasoning, the problem is how to feature the robustness that is necessary for speech understanding. This paper addresses the issue of the robustness of the Data Oriented Parsing (DOP) model within a Dutch speech-based dialogue system. It presents an extension of the DOP model into a head-driven variant, which allows for Markovian generation of parse-trees. It is shown empirically that the new variant improves over the original DOP model on two tasks: the formal understanding of speech utterances, and the extraction of semantic-concepts from "word-lattices" output by a speech-recognizer.

## 1 Introduction

Speech understanding is a challenging task for probabilistic parsing models. The problem with speech utterances is that they do not always abide by linguistic grammar rules. Speech utterances exhibit phenomena such as repairs, repetitions and hesitations, all of which are considered problems outside the domain of linguistic research. The challenge for a parsing model is to deal with such phenomena. A greater challenge is set by real speech understanding tasks in noisy environments, such as speech over the telephone. In such cases, the speech-recognizer's accuracy degrades and language models might be of some use in recovering some of the lost accuracy.

OVIS is a national project of the Dutch Organization for Scientific Research (NWO) aiming at building a prototype dialogue system for the domain of railway time-table information. The dialogue in OVIS takes place over the telephone. The system interacts through "dialogue" with a human user aiming at providing the user with travel information. The system consists of different modules including a dialogue manager, a speech recognizer, a Natural Language Processing (NLP) module, and a language generation module. In speech understanding, we focus on the role of the NLP module which consist of the interface between the speech recognizer and the dialogue manager. The output of the speech recognizer is processed by the NLP module, and the "semantic content" of the user's utterance is extracted and supplied to the dialogue manager.

The OVIS system provides an interesting problem for language modeling because it addresses a real application of the processing of spoken language in a noisy environment. Furthermore, the task of language understanding in OVIS has been formalized in terms of *domain dependent semantic criteria* making the evaluation of language models more linked to the actual task. In earlier work, two language

models were compared on this task [22, 21]: a system based on a broad-coverage grammar for Dutch, and a system based on the Data Oriented Parsing (DOP) model.

In this paper we address the problem of robust language understanding within the OVIS domain using the DOP model. We present a new version of the DOP model which is more suitable for the processing of spoken language utterances than the original DOP model. Robustness in this new version, called the Tree-gram model, is the result of integrating into DOP the "Markovian" approach for grammar-rule generation, as in some exiting models, e.g. [10, 8]. We exhibit significant empirical improvements, over the DOP model, in both OVIS tasks: (1) the formal understanding of spoken utterances and (2) the extraction of the "best" semantic content from an ambiguous word-lattice (also called "word-graph"), output by a speech-recognizer.

The structure of this paper is as follows. Section 2 provides a short overview of the OVIS system, the OVIS tree-bank and the experience with applying DOP within OVIS. Section 3 provides a review of the DOP model and presents the new version: the Tree-gram model. Section 4 attempts a theoretical comparison between the two models concerning the issue of robustness. Section 5 exhibits the empirical results of experiments in applying DOP and the Tree-gram model to speech understanding within the OVIS domain. Finally, section 6 concludes the paper.

## 2 Brief overview of OVIS

In the OVIS demonstrator system, the communication with the human user takes place over the telephone through a spoken-language dialogue aiming at providing the user with travel information. The dialogue manager in OVIS maintains an "information state" to keep track of the information extracted from the user's answers to questions posed by the system. This information state consists of a small number of slots that are typical of train travel information, e.g. origin, destination, date, time. The semantic content of a user's utterance is used for updating the slots in the information state. Hence, the output of the natural language processing module is exactly an "update expression" specifying what slots must be updated and with what values. In OVIS, these update expressions are terms in a formal language of "update semantics" developed by [23]. This update language provides ways for expressing various updates including "speech-act information" such as denials and corrections. Here, we are merely interested in the fact that the update-language has been expressed in terms of a formally specified hierarchy of the slots: for example, the slots "place" and "time" provide more specific information over the slot "destination".

The OVIS tree-bank [13] contains 10000 utterances annotated syntactically and semantically. The interesting part of the OVIS tree-bank is that the semantics is largely compositional [5]: the semantics of a non-terminal node is expressed in terms of the semantics of its child-nodes; this is expressed as a simplified form of Lambda expressions, e.g. "(D1;D3)" where D$i$ refers to the $i^{th}$ child. The part-of-speech (POS) tag labeled nodes are annotated with ground semantic expressions, e.g. "(PPN-amsterdam amsterdam)" or "(PP-origin.place naar)". In [5] a method is also described for transforming the semantic expressions at every node into a label using the semantic hierarchy of [23]: roughly speaking, the semantic expressions are categorized according to the kind of slots which they aim at filling, e.g. place expressions specify a category while time expressions specify another, different category. Crucially, this semantic categorization aims at labeling the grammar rules in the tree-bank trees in such a way that it is possible to retain the exact semantics of the tree-bank trees unambiguously. In this work we employ the OVIS tree-bank enriched with this categorization scheme.

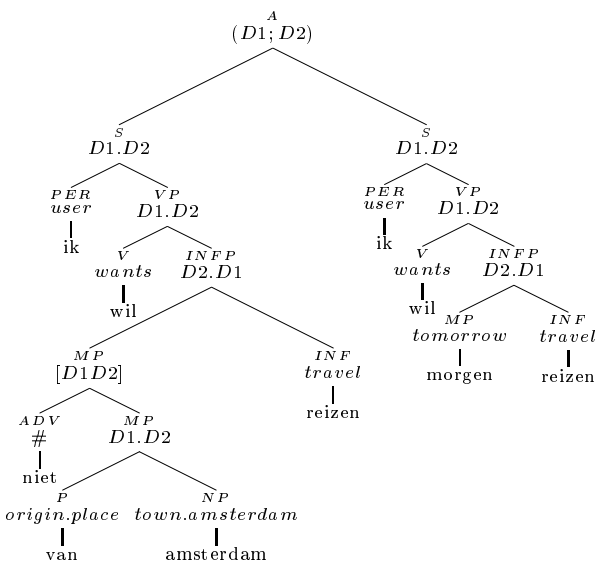


Figure 1: An example OVIS parse-tree.

Figure 2 exhibits an OVIS parse-tree with the compositional semantics shown under the label of every node. The update expression of the whole utterance is computed compositionally in a bottom-up fashion, substituting for $Di$ the update expression of the $i^{th}$ child. The English equivalent of the given utterance is "*I do not want to travel from Amsterdam I want to travel tomorrow*". The update expression for this particular parse-tree is: *(user.wants.travel.[#origin.place.town.amsterdam]; user.wants.travel.tomorrow)*, where the operator "[#A]" denotes the denial of A, "A;B" denotes the concatenation of update expressions A and B and "A.B" denotes that B is a more specific slot than A or that B is the update value for slot A

The present paper addresses the problem of applying the Data Oriented Parsing (DOP) model [15, 2] to the understanding of utterances and word-graphs that are output by a speech recognizer [14] in the OVIS domain. In earlier experiments [22], the DOP model scored significantly worse than a complex hybrid system which combines a broad coverage grammar for Dutch, a word trigram model and a smart concept spotting strategy [21]. Our research revealed three sources of problems with DOP: *lack of robustness, weak lexicalization and a biased probability estimation method.* Among these three problems, the focus here is on robustness.

In [18] we extend the DOP model with capabilities similar to the so called Markov-Grammar models e.g. [10, 8]. The new model, called the Tree-gram model, is capable of generating parse-trees that the original DOP model is not capable of generating, possibly enhancing robustness. Furthermore, the model allows for head-driven parsing, albeit the implementation described in [18] is not head-lexicalized in the sense argued for by e.g. [10, 8]. Hence, the problem with the Tree-gram model, just like DOP, is that it does not condition the model parameters on lexical information, i.e. word-occurrence. The question is whether this kind of "weakly lexicalized" model constitutes any improvement on the original DOP model ? Next we show that the Tree-gram model significantly improves on the results of the DOP model, in parsing and interpretation of speech utterances as well as word-graphs. We show that on parsing and interpretation of speech utterances, the model achieves results that come very close to those exhibited by the Dutch broad coverage grammar on the same task. Despite these

encouraging results, we conclude this study with stressing the weakness of the unlexicalized nature of the DOP model and the current unlexicalized implementation of the Tree-gram model, and speculate on future work in this direction.

# 3 Overview of the DOP and Tree-gram models

A probabilistic model assigns a probability to every parse-tree given an input sentence $S$, thereby distinguishing one parse

$$\begin{aligned} T^* &= argmax_T\ P(T|S) \\ &= argmax_T\ \frac{P(T,S)}{P(S)} = argmax_T\ P(T,S). \end{aligned}$$

The probability $P(T,S)$ is usually estimated from co-occurrence statistics of linguistic phenomena extracted from a given tree-bank. In generative models, the tree $T$ is generated through top down derivations that rewrite the start symbol $TOP$ into the sentence $S$. Each rewrite-step involves a "rewrite-event" together with its estimated probability of application. Next we provide a short overview of two generative models: the DOP model and the Tree-gram model.

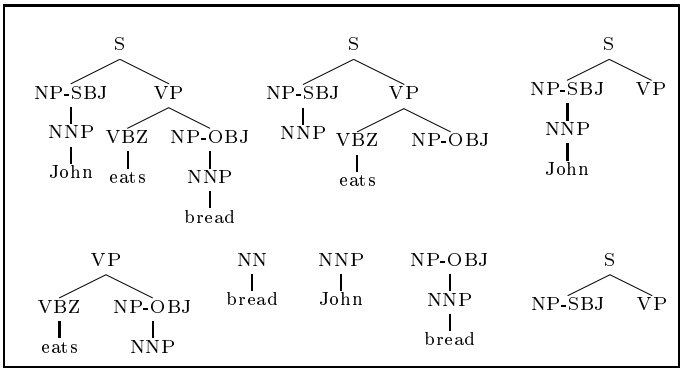


Figure 2: Some subtrees: DOP decomposition.

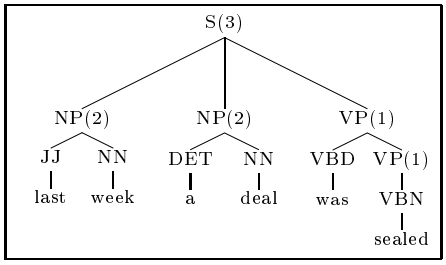


Figure 3: An example parse-tree. Between brackets node head-child numbers.

## 3.1 The DOP model

In Data-Oriented Parsing (DOP1) e.g. [2], the rewrite-events are "subtrees" of the tree-bank trees: a subtree of a given parse-tree is a *multi-node connected subgraph in which every node either dominates all its children*[1] or it dominates none of them. If we view the tree-bank trees as generated by derivations of some linguistic Context-Free Grammar (CFG) [1], then a DOP subtree consists of one or more connected CFG rules that co-occur in a tree-bank tree. Hence, the parse-trees and sentences which a DOP model recognizes are exactly those which the original linguistic CFG does. The difference, however, is in the fact that DOP assigns probabilities to the subtrees, which can be seen as probabilities of co-occurrences of CFG rules. Figure 2 shows some DOP subtrees extracted from the parse-tree that is identical to the subtree in the top-left corner of the same figure.

The probability of a subtree in DOP is estimated from its relative frequency in the tree-bank. Let $root(t)$ denote the root-label of the root of any subtree $t$, and let $freq(x)$ denote the frequency-count of subtree $x$ in the tree-bank. The probability of a subtree $t$ is estimated by the formula:

[1] Thereby preserving the direct dominance relations of the original tree-bank parse-trees.

$P(t|root(t)) = \frac{freq(t)}{\sum_{x:root(x)\equiv root(t)} freq(x)}$. The probability of a derivation $d$, involving subtrees $t_1 \cdots t_n$, is estimated as

$$P(d) = \prod_{1 \leq i \leq n} P(t_i|root(t_i)).$$

According to Bod [3, 4, 2], the probability of a parse-tree $T$ and a sentence $S$, generated respectively by the sets of derivations $D(T)$ and $D(S)$, are estimated by $P(T,S) = \sum_{d \in D(T)} P(d)$, and by $P(S) = \sum_{d \in D(S)} P(d)$. In [16] it is shown that the problems of disambiguation under the DOP model, concerning the computation of the Most Probable Parse (MPP) (and the Most Probable Sentence (MPS) in a word-graph) are NP-complete, i.e. it is not possible to devise deterministic polynomial algorithms to exactly compute the MPP (or MPS from a word-graph). Here we suggest to approximate the probabilities of a parse-tree $T$ and a sentence $S$ as follows: $P(T) \approx argmax_{d \in D(T)} P(d)$, $P(S) \approx argmax_{d \in D(S)} P(d)$. This formulation has the advantage of being efficiently solvable by a polynomial-time algorithm, similar to the well known Viterbi-algorithm [24]. The negative side, however, is that it still contains some of the bias that the original DOP definition had (see [6]) and that it under-estimates the probabilities. However, we think that this under estimation in itself is not harmful since the exact values are not important as much as the relative ordering between the parses (and sentences). We suspect that under some assumptions that has to do with the nature of the given tree-bank annotation, the relative frequency of DOP subtrees, as in e.g. Stochastic CFGs (SCFGs), provides a "useful" ordering over the derivation probabilities. Since this is not the place to elaborate on this theoretical point, we seek the help of empirical evidence on this issue, as exhibited in the experiments in section 5.

### 3.2 The Tree-gram model

In the Tree-gram model, the set of "rewrite-rules" subsumes the CFG rules and the connected combinations thereof that can be extracted from the tree-bank, i.e. the DOP subtrees. We refer to these rewrite-events with the term *Tree-grams* (abbreviated T-grams). A T-gram extracted from a parse-tree in the training tree-bank is a *multi-node connected subgraph* of that parse-tree. Note that the set of T-grams extracted from a tree-bank subsumes (or is equal to) the set of DOP subtrees extracted from the same tree-bank; the set of T-grams includes connected subgraphs of the training parse-trees which do not retain the direct dominance relation (i.e. parent-child) as found in the tree-bank. Hence, when extracting a T-gram from some node $\mu$, not necessarily all children of $\mu$ are included into the T-gram. In the current implementation, however, we demand that the children of $\mu$ that are included in the T-gram are direct sisters to one another, e.g. we do not allow including the first and the fifth child if any of the second, third and fourth are not included also. This simplifies the parsing algorithms. Some example Tree-grams extracted from the parse-tree in figure 3 are shown in Figure 4.

T-grams are inspired by Markov Grammars [10, 8]: in fact T-grams provide a direct general-form both for Markov Grammar rules (called bilexical dependencies) as well as DOP subtrees. Next we describe in short how T-grams are employed in the Tree-gram model. Further formal detail can be found in [18].

We assume that for every non-leaf node $\mu$ in the training tree-bank trees, one of its child nodes is specified as being the "head-child": the child that dominates the head-word of $\mu$. The Tree-grams acquired from the tree-bank trees are partitioned into three subsets, called *roles*, according to the kind of children that the root of a Tree-gram dominates. When a Tree-gram's root dominates its head-child

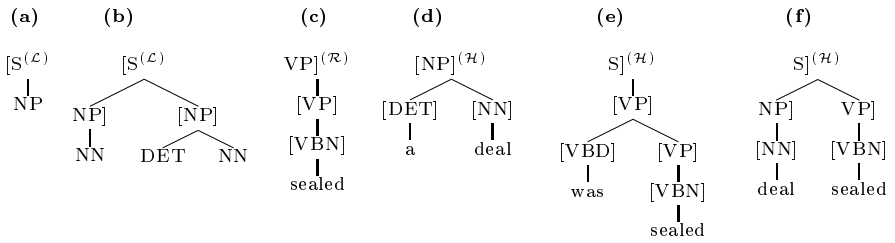


Figure 4: Some T-grams extracted from the tree in figure 3: the superscript on the root label specifies the *T-gram role*,. e.g. the left-most T-gram is in the LEFT role. Non-leaf nodes are marked with "[" (left-STOP) and "]" (right-STOP) to specify whether they are complete from the left/right or both (the other non-complete nodes, i.e. from both sides, are not marked at all).

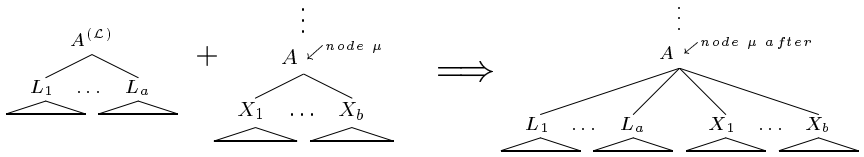


Figure 5: A T-gram is generated by attachment at $\mu$ in a partial parse-tree. The T-gram being generated is marked with $\mathcal{L}$ (or $\mathcal{R}$) to denote its role. We show a LEFT Tree-gram being generated.

(and possibly other children), the Tree-gram is in the "Head" role; when it dominates only children which are originally found (in the tree-bank tree from which the Tree-gram was extracted) to the left (right) of the head-child (e.g. left-modifiers of the head-child), it is in the LEFT (RIGHT) role. In essence, these roles express information about the nature of the Tree-gram with respect to the context from which it was extracted. Some example Tree-grams are shown in Figure 4.

In contrast with DOP subtrees, Tree-grams allow also for a "horizontal" expansion of the parse-trees as depicted in figure 5. This horizontal expansion of parse-trees takes place by combining Tree-grams labeled with the same root node in a "Markovian fashion". Formally speaking, the horizontal combination of Tree-grams must have a probability of terminating. Therefore, the horizontal combination of Tree-grams is governed by a formal definition of when a node "terminates". The termination process is "inherited" from the tree-bank trees: the sequence of children of every node in a tree-bank tree is explicitly marked as terminated from the left and from the right by a special symbol "STOP". To the left of the sequence of children, the STOP is denoted by "[" and to the right it is denoted by "]". When a Tree-gram is extracted from a tree in the tree-bank, the STOP symbols ("[" and "]") might either be included or they might not be included with the non-leaf nodes of the Tree-gram. For any non-leaf node in a Tree-gram, if both STOP symbols are included along with its children, the node is called *complete*. When STOP is absent from either the left or right hand sides of a node (or both), the node is incomplete. In the latter case, the partial parse-trees that the node dominates may be extended with additional Tree-grams as described next (hence, non-terminal leaf nodes are always incomplete allowing substitution as in DOP). See figure 4 for examples.

Tree-gram rewrite processes, i.e. derivations, start from the start-symbol TOP, which is an incomplete non-terminal. At each rewrite-step, an incomplete node $\mu$ is selected and rewritten by a suitable Tree-gram as follows. When $\mu$ is a leaf node labeled with a non-terminal $A$, it is rewritten by a HEAD Tree-gram with a root labeled $A$ (much like rewriting takes place in DOP, i.e. "vertical

expansion"); when a non-leaf node $\mu$ is labeled with a non-terminal $A$ and it is incomplete, it may be rewritten with LEFT and RIGHT Tree-grams that have roots also labeled $A$. The latter rewriting allows *horizontal* expansion of the parse-tree at node $\mu$ (see Figure 5). The rewrite process terminates when the resulting parse-tree consists entirely of complete nodes.

The conditioning context (or history) $h_t$ in the probability $P(t|h_t)$ of a Tree-gram $t$, consists of the label of the root-node of $t$ (i.e. $root(t)$), the role of $t$ (i.e. HEAD, LEFT or RIGHT) and the following specific information:

- A HEAD Tree-gram's probability is further conditioned on the POS tag of the head-word of the root node of $t$.
- A LEFT (RIGHT) Tree-gram's probability is further conditioned on
  - the label of the head-sister of its root-node, and on
  - the label of the sister to the right (resp. left) of the root-node in the original tree-bank tree, thereby yielding a $1^{st}$*-order Markovian process.*

Our conditioning context is similar to those used in e.g. [8]. Just like in DOP, the probability of a Tree-gram derivation $d$ involving the sequence of T-grams $t_1, \cdots, t_n$ is estimated by

$$P(d) = \prod_{i=1}^{n} P(t_i|h_{t_i}).$$

As we argued for DOP earlier, we approximate the probabilities of a parse-tree and a sentence as the highest probability of any of their derivations.

# 4 A theoretical comparison on robustness

Formally speaking, both models assign probabilities to context-free languages. However, as Bod shows [2], the probability distribution assigned to the members of the set of parse-trees generated by a DOP model cannot always be generated by a Stochastic Context-Free Grammar (SCFG) [11]. For a given tree-bank, the set of sentences accepted (or generated) by the Tree-gram model acquired from that tree-bank is a superset of (or equal to) the set accepted by the DOP model acquired from the same tree-bank. The same relation applies between the respective sets of parse-trees generated by both models. Hence, the Tree-gram model, just like "Markov-grammars", generates sentences and parse-trees that cannot be generated the DOP model or by the linguistic grammar that underlies the annotation of the tree-bank.

The question of course is: how do the distributions over derivations, parse-trees and sentences generated by both models compare to one another ? The empirical estimation of probabilities from a tree-bank makes this comparison complicated due to problems of bias in the current method of probability estimation which both models suffer from [6]. However, if we restrict the subtrees and Tree-grams that are acquired from the tree-bank by formal means (i.e. restrictions on depth or width of a subtree/Tree-gram – see section 5), then one can already sense that both models will assign a "similar" relative ordering to the derivations that they generate. This can be seen only intuitively by the fact that all subtrees of the DOP model are included as HEAD Tree-grams in the Tree-gram model with the same relative frequency (up to a finer conditioning context in Tree-grams). In any case, the Tree-gram model, at least theoretically, allows for a more "subtle" model than the DOP

| system | Match | Prec. | Recall |
|---|---|---|---|
| DOP | 93.0 | 94.0 | 92.5 |
| **Tgram** | **94.5** | **95.0** | **95.6** |
| DBCG | 95.7 | 95.7 | 96.4 |

Table 1: Results on utterances

| system | WA | SA | Match | Prec. | Recall |
|---|---|---|---|---|---|
| DOP | 72.2 | 71.8 | 77.2 | 82.0 | 77.3 |
| **Tgram** | **79.6** | **74.0** | **81.4** | **85.2** | **84.3** |
| Tgram - DOP | +7.4 | +2.2 | +4.2 | +3.2 | +7.0 |

Table 2: Results on word-graphs

model. The question is of course, does this subtlety of the Tree-gram model translate into more robust processing for speech understanding ? We investigate this question in the context of the OVIS speech-understanding system in an empirical way in the next section.

# 5 Parsing and interpretation of the OVIS domain

Before applying the Tree-gram model to the OVIS domain, it is necessary to specify how we signify the head-children in the tree-bank trees. Given the compositional semantics in the tree-bank, with a few exceptions, every CFG rule has a semantic formula associated with it. This formula expresses how the semantics of the node is composed from the semantics of its children using a small set of operators, e.g. concatenation "D1;D2", correction "[!D2]", denial "[#D3]". Some of these operators take a single argument, others take two arguments. We decided to specify, the head-children using these formulae through a few rules of thumb, (e.g. take the first child specified in the formula, except for a few specific situations). This specifies the head-children for all tree-bank nodes unambiguously.

The Markovian nature of the Tree-gram model, allows us to apply the Katz backoff smoothing technique [12, 9] using the $0^{th}$-order Markovian conditioning for LEFT and RIGHT Tree-grams: we apply that to all T-grams of depth 1 only. Furthermore, we allow backoff on the stop symbols "[" and "]" on the root-node of a T-gram of depth 1 in one of two ways: (1) we add a stop symbol "[" to the left ("]" to the right) of the node with a suitable backoff probability, or (2) we remove these symbols, if they are there, with a suitable backoff probability. The resulting "backoff T-grams" are included in the model together with the original ones. For semantic interpretation, all new rules generated by the Tree-gram model are assigned a heuristic formula depending on the parse-tree in which they occur; the heuristic semantics of a new rule depends on the types of the semantics of the child-nodes, and aims at combining these types in acceptable ways (with respect to the OVIS update language).

We use the same parser for the DOP model as well as the Tree-gram model [17]. This is a CYK [25] based algorithm using an optimized version of the Viterbi-algorithm with a simple pruning technique. The parser is applicable to utterances as well as word-graphs (the latter extension is straightforward - see [19, 17]).

We trained a DOP model (with subtree depth[2] upperbound 4) and a Tree-gram model (with Tree-gram depth[3] upperbound 5) on the same training tree-bank of 10000 utterances. We compare the models on a held-out set of 1000 utterances, which was used for similar experiments in [22]. We also report preliminary results on a set of 500 speech-recognizer's word-graphs[4].

[2] In various experiments reported in [5, 22, 17], it turns out that DOP models with subtrees deeper than 4 show worse results than a DOP model with subtree depth upperbound 4.

[3] Tree-gram depth here is measured after "binarization" of the Tree-gram in a head-driven fashion. This head-driven "binarization" transforms the children of every node as follows: the children to the left (right) of the head-child are transformed into a left (resp. right) branching binary tree (the head-child remains directly under the current node). After this process every node dominates at most three children (head-child and a left and right nodes added by the process). Hence, Tree-gram depth is a mix of actual depth with the branching factor under the internal nodes.

[4] The probability of a DOP/Tree-gram derivation of a path in an input word-graph is multiplied with the speech-

The semantic evaluation criteria have been developed by [20] following similar criteria suggested in [7]. A semantic expression is translated into a set of "semantic units"; each semantic unit addresses a specific OVIS slot. Given this view on semantic expressions, now we can compare the semantic-expression $U$ output by a given system to the gold-standard expression $G$ in the same way as in Labeled Recall and Precision in syntactic parsing: (1) semantic exact match is the average test-set utterances for which $U \equiv G$, (2) semantic recall (precision) is the average, over the test-set utterances, of $\frac{|U \cap G|}{|G|}$ (resp. $\frac{|U \cap G|}{|U|}$ – when $|U| = 0$, this is by definition zero). For word-graph parsing we also use the word-accuracy (WA) and sentence-accuracy (SA) measures to compare the proposed utterance $P$ to the gold $G$: $WA = 1 - \frac{d}{n}$, where $n$ is the length of the $G$, and $d$ is the Levenshtein distance between $G$ and $P$ (see [22] for detail).

Table 1 shows the results of the Tree-gram model, the DOP model and the Dutch broad-coverage grammar (DBCG) on utterances. Clearly, the latter system is still producing the best results, however the Tree-gram model has narrowed the gap on utterances for recall from 3.9% (DOP) to 0.8% (Tgram). For word-graphs, our results can not be compared to those of the DBCG-based system (although we suspect that the DBCG improves over the Tree-gram results) because this preliminary experiment is on a different set than the final test-set. However, the Tree-gram model improves over DOP by at least 7% on WA and semantic recall and 4.2% on semantic match.

Our explanation to the improvements on DOP's results is that on utterances the Tree-gram model is capable of producing parses which DOP cannot produce; on about 2.2% of the utterances, DOP does not produce any parse and these utterances are usually some of the longer ones. Then, in a few more cases, it seems that DOP produces only a less useful parse than the Tree-gram model. When we inspected some of the cases it turned out that DOP tends to assign the special label "ERROR" (used to mark repetitions and corrections in the OVIS tree-bank) to various constituents for which it could not find an approximate label.

We can think of various reasons why the Tree-gram's results are still lagging behind those of the DBCG results: (1) the DBCG grammar has been developed manually in an incremental fashion inspecting how the system behaves on a large collection of over 100000 utterances from the OVIS domain, while our models are trained on a relatively small training tree-bank of 10000 parse-trees, (2) the tree-bank syntactic and semantic annotations contain minor inconsistencies that disturb the models, (3) the model probabilities are not conditioned on lexical information, and (4) in contrast to the DBCG module, we did not try to transform "informative", yet formally wrong, semantic formulae output by the parser[5].

# 6 Conclusions

We have shown how the DOP model can be extended in a useful way for more robust parsing of speech utterances. The present extension, called the Tree-gram model, generalizes over the DOP model by assigning non-zero probability values to some utterances for which a DOP model assigns probability zero. We are encouraged by the fact that the Tree-gram model has narrowed the gap with the results

recognizer's likelihoods that are found on the transitions in the path (after applying a simple scaling heuristic). This is a kind of standard Bayesian combination of the two modules.

[5] It is possible to devise a few set of heuristic rules based on the OVIS semantic hierarchy, for the correction of such "informative" formulae: these formulae usually consist of multiple correct subformulae without the formally necessary combination operators. Often the operators can be guessed by inspection of the semantic types of the subformulae and the OVIS semantic hierarchy.

of the Dutch Broad-Coverage Grammar (DBCG) system: our system is automatically acquired from a tree-bank, while the DBCG took more than three years to develop. Nevertheless, it remains not clear how fast each of the two systems can be successfully adapted to a new domain of language use.

Our preliminary results on word-graphs improve considerably over DOP's results. Again this is due to the more robust nature of the new model in comparison with the original DOP model. However, in first inspection of some of the problems, we find that the model still suffers from the weak lexicalization, just like DOP. The gain is solely due to the fact that the Tree-gram model could parse many more of the word-graph paths than DOP did, thereby having more paths to choose from.

We suspect that the fact that our word accuracy and sentence accuracy are still lagging behind those of simpler models (e.g. a word trigram model) implies that neither DOP nor the current Tree-gram model is sufficiently suitable for the task of speech-understanding from word-graphs. As the word-graphs become larger, it becomes harder to select the correct sequence of words. In such cases, word co-occurrence probabilities are at least as important as probabilities that express "grammatical plausibility", which is taken care of by the current models. We think that it is necessary to condition the probabilities in these models on lexical information, possibly in a head-driven fashion similar to the bilexical dependency models, e.g. [10, 8].

## Acknowledgements

This work is funded by a project of the Netherlands Organization for Scientific Research (NWO). I am grateful to Remko Scha, Remko Bonnema, and Walter Daelemans for discussions and support. Some software packages, which were used for evaluation of the models, were developed by Gertjan van Noord and Remko Bonnema. The preparation of the final version of this paper was facilitated by hardware and internet connection by Telser Cabinas Internet (Cusco, Peru) and much understanding from Vincent, Marije and Didi.

## References

[1] A. Aho and J. Ullman. *The Theory of Parsing, Translation and Compiling*, volume I, II. Prentice-Hall Series in Automatic Computation, 1972.

[2] R. Bod. *Enriching Linguistics with Statistics: Performance models of Natural Language*. PhD thesis, ILLC-dissertation series 1995-14, University of Amsterdam, 1995.

[3] Rens Bod. Monte Carlo Parsing. In *Proceedings Third International Workshop on Parsing Technologies, Tilburg/Durbuy*, 1993.

[4] Rens Bod. The Problem of Computing the Most Probable Tree in Data-Oriented Parsing and Stochastic Tree Grammars. In *Proceedings Seventh Conference of The European Chapter of the ACL, Dublin*, March 1995.

[5] R. Bonnema, R. Bod, and R. Scha. A DOP Model for Semantic Interpretation. In *Proceedings of ACL-97*, Madrid, Spain, July 1997.

[6] R. Bonnema, P. Buying, and R. Scha. A new probability model for data oriented parsing. In Paul Dekker and Gwen Kerdiles, editors, *Proceedings of the 12th Amsterdam Colloquium*, Amsterdam,

The Netherlands, december 1999. Institute for Logic, Language and Computation, Department of Philosophy.

[7] M. Boros, W. Eckert, F. Gallwitz, G. Gorz, G. Hanrieder, and H. Niemann. Towards understanding spontaneous speech: Word accuracy vs. concept accuracy. In *Proceedings of the Fourth International Conference on Spokenm Language Processing (ICSLP 96)*, Philadelphia, 1996.

[8] E. Charniak. A maximum-entropy-inspired parser. In *Report CS-99-12*, Providence, Rhode Island, 1999.

[9] S. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Technical report TR-10-98*, Harvard University, August 1998.

[10] M. Collins. Three generative, lexicalized models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the EACL*, pages 16–23, Madrid, Spain, 1997.

[11] F. Jelinek, J.D. Lafferty, and R.L. Mercer. *Basic Methods of Probabilistic Context Free Grammars, Technical Report IBM RC 16374 (#72684).* Yorktown Heights, 1990.

[12] S.M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3), 1987.

[13] K. Sima'an / R. Scha, R. Bonnema, and R. Bod. *Disambiguation and Interpretation of Word-graphs using Data Oriented Parsing.* Probabilistic Natural Language Processing in the NWO priority Programme on Language and Speech Technology, Amsterdam, November 1996.

[14] M. Oeder and H. Ney. Word graphs: An efficient interface between continuous-speech recognition and language understanding. In *ICASSP Volume 2*, pages 119–122, 1993.

[15] R. Scha. Language Theory and Language Technology; Competence and Performance. In Q.A.M. de Kort and G.L.J. Leerdam, editors, *Computertoepassingen in de Neerlandistiek*, Almere: LVVN-jaarboek (can be obtained from http://www.hum.uva.nl/computerlinguistiek /scha/IAAA/rs/cv.html#Linguistics), 1990.

[16] K. Sima'an. Computational Complexity of Probabilistic Disambiguation by means of Tree Grammars. In *Proceedings of COLING'96*, volume 2, pages 1175–1180, Copenhagen, Denmark, August 1996.

[17] K. Sima'an. *Learning Efficient Disambiguation.* A PhD dissertation. ILLC dissertation series 1999-02 (Utrecht University / University of Amsterdam), Amsterdam, March 1999.

[18] K. Sima'an. Tree-gram Parsing: Lexical Dependencies and Structual Relations. In *Proceedings of the* $38^{th}$ *Annual Meeting of the Association for Computational Linguistics (ACL'00)*, pages 53–60, Hong Kong, China, 2000.

[19] G.J. van Noord. The intersection of finite state automata and definite clause grammars. In *Proceedings of ACL-95*, 1995.

[20] G.J. van Noord. Evaluation of OVIS2 NLP components. In *Technical Report #46, NWO Priority Programme Language and Speech Technology*, 1997.

[21] G.J. van Noord, G. Bouma, R. Koeling, and MJ Nederhof. Robust Grammatical Analysis for spoken dialogue systems. *Journal of Natural Language Engineering*, 5 (1):45–93, 1999.

[22] G. Veldhuijzen van Zanten, G. Bouma, K. Sima'an, G.J. van Noord, and R. Bonnema. Evaluation of the NLP Components of the OVIS2 Spoken Dialogue System. In I. Schuurman F. van Einde and N. Schelkens, editors, *Proceedings of* Computational Linguistics In the Netherlands 1998, 1999.

[23] Gert Veldhuijzen van Zanten. *Semantics of update expressions.* Technical report 24, NWO Priority Programme Language and Speech Technology, *http* : *//odur.let.rug.nl* : 4321/, 1996.

[24] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Information Theory*, IT-13:260–269, 1967.

[25] D.H. Younger. Recognition and parsing of context-free languages in time $n^3$. *Inf.Control*, 10(2):189–208, 1967.