Available online at www.sciencedirect.com

SciVerse ScienceDirect

Speech Communication 54 (2012) 715–731

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Syntactic language modeling with formal grammars

Tobias Kaufmann *, Beat Pfister

Speech Processing Group, Computer Engineering and Networks Laboratory, ETH Zurich, Gloriastrasse 35, 8092 Zurich, Switzerland

Received 4 March 2010; received in revised form 29 December 2011; accepted 3 January 2012
Available online 12 January 2012

Abstract

It has repeatedly been demonstrated that automatic speech recognition can benefit from syntactic information. However, virtually all syntactic language models for large-vocabulary continuous speech recognition are based on statistical parsers. In this paper, we investigate the use of a formal grammar as a source of syntactic information. We describe a novel approach to integrating formal grammars into speech recognition and evaluate it in a series of experiments. For a German broadcast news transcription task, the approach was found to reduce the word error rate by 9.7% (relative) compared to a competitive baseline speech recognizer. We provide an extensive discussion on various aspects of the approach, including the contribution of different kinds of information, the development of a precise formal grammar and the acquisition of lexical information.

Keywords: Large-vocabulary continuous speech recognition; Language modeling; Formal grammar; Discriminative reranking

1. Introduction

Acoustic models in automatic speech recognition have difficulty in distinguishing between alternative transcriptions that are phonetically similar. The role of a language model is to complement the acoustic model with prior information about the word sequences that occur in a particular language.

The most widely used class of language models, the so-called n-grams, is based on the assumption that the probability of a word only depends on a small number of preceding words. These models can easily be trained on large amounts of unannotated text, and they perform extremely well despite of their simplicity.

However, n-grams fail to capture many dependencies that are present in natural language. In our experimental data, the n-gram language model prefers the incorrect transcription "*einer, der sich für die Umwelt einsetzen*" (translation of the correct transcription: *someone who stands up for the environment*). This transcription is ungrammatical because the number of the relative pronoun *der* (singular) does not agree with the number of the finite verb *einsetzen* (plural). In order to capture this instance of subject–verb agreement, an n-gram would have to consider at least five words preceding the verb. Moreover, the training corpus would have to contain the correct word sequence "*der sich für die Umwelt einsetzt*" (*who stands up for the environment*), which is quite unlikely. We do not claim that n-grams and variants such as skip n-grams or class-based n-grams cannot possibly deal with this particular example. But it is easy to find many similar examples for any language model that does not properly capture the dependencies of natural language.

In the last decade, progress in the field of statistical parsing has led to the development of more powerful language models that also incorporate syntactic structure. These language models have been shown to increase recognition accuracy for broad-domain speech recognition tasks.

Unlike statistical parsers, formal grammars are designed to accurately distinguish between grammatical and ungrammatical sentences. Formal grammars have been successfully applied to narrow-domain natural language understanding tasks, but not to broad-domain speech

* Corresponding author.
E-mail address: tobias.kaufmann@stairwell.ch (T. Kaufmann).

0167-6393/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2012.01.001

recognition. This may partly be due to the difficulty of scaling such approaches to broad domains. Precise broad-domain grammars are notorious for their lack of coverage and at the same time fail to exclude many incorrect but nevertheless grammatical transcriptions. This greatly diminishes the value of grammaticality information. In addition, the development of grammars and lexica for broad domains is a difficult and laborious task. All these effects can be controlled much more easily for restricted domains.

This situation is somewhat unsatisfying as formal grammars do have properties that are interesting for language modeling. They appear to be the most adequate means to formalize hard linguistic constraints. If they are complemented with statistical models for disambiguation, these models typically require relatively few syntactically annotated training data compared to statistical parsers. And finally, formal grammars encode linguistic constraints that are – at least to some degree – orthogonal to word-based statistical models, including n-grams and statistical parsing models.

In this paper, we present a novel approach of integrating formal grammars into a statistical speech recognition framework. We show that this approach can significantly improve the accuracy of a competitive speech recognizer on a broad-domain task. We further report on a series of experiments which reveal different characteristics of the proposed approach and give an idea of how much syntactic information can contribute to speech recognition. In addition to these experiments, we will describe the underlying linguistic resources and explain the formalisms and methods that were used to develop these resources.

This article is a condensed version of the main author's Ph.D. thesis (Kaufmann, 2009). Some of the experimental results have previously been published in (Kaufmann et al., 2009).

1.1. Statistical parsers as language models

Virtually all syntactic language models for large-vocabulary continuous speech recognition are based on statistical parsers.[1] A statistical parser basically models the relation between a word sequence W and a parse tree T as a probability distribution $P(T, W)$ or $P(T|W)$. The parameters of this distribution are estimated on large syntactically annotated text corpora, so-called *treebanks*.

A common way of integrating statistical parsers into speech recognition is to directly use them as a generative language model $P(W) = \sum_T P(T, W)$. This kind of approach was pursued by Chelba and Jelinek (2000), Roark (2001b), Charniak (2001), Xu et al. (2002), Wang and Harper (2003). More recently, Collins et al. (2005) introduced a discriminative rather than generative approach to syntactic language modeling. In their setup, a statistical parser determines the most likely parse tree $T^\star = \text{argmax}_T P(T, W) = \text{argmax}_T P(T|W)$ for each recognition hypothesis W. A discriminative statistical model then chooses the most likely recognition hypothesis based on a set of features that were extracted from the word sequences and their respective parse trees. A similar approach was followed by McNeill et al. (2006).

Syntactic language models based on statistical parsers offer several advantages. They can consider the dependencies between actual word forms and thus are able to capture semantics and fixed expressions in a similar way to n-grams. For example, they may prefer "*he gave a talk*" to "*he gave a walk*" if the words *gave* and *talk* occur in verb-object relation more frequently than *gave* and *walk*. Further, most statistical parsers employ a sufficiently productive (often infinite) set of grammar rules that allows them to produce a parse tree for any word sequence W. This robustness property is important for dealing with hypotheses that are ungrammatical or whose syntax is not covered by the treebank.

Statistical parsers have been shown to improve speech recognition on broad-domain tasks such as transcribing spontaneous telephone conversations (Chelba and Jelinek, 2000; Wang et al., 2004; Collins et al., 2005; McNeill et al., 2006), broadcast news shows (Chelba and Jelinek, 2000) or read newspaper articles (Chelba and Jelinek, 2000; Roark, 2001b; Wang and Harper, 2002; Xu et al., 2002 and Hall and Johnson, 2003).

1.2. Formal grammars

It might be appropriate to first explain which particular kinds of formal grammar we are concerned with. A prototypical formal grammar is intended to discriminate between those word sequences that are grammatical (i.e. that can be generated by a sequence of rule applications) and those which are not. In natural language processing and computation linguistics, a grammar is typically used to approximate a natural language rather than define an artificial language. In this context, we informally call a grammar *precise* if it accurately predicts whether sentences are grammatical with respect to some natural language or not.

Predicting the grammaticality of a sentence is generally not sufficient for applications of natural language processing. Thus, we make a few additional assumptions:

- The grammar allows to determine the possible syntactic structures of a grammatical sentence. Based on these structures, statistical models can compute probabilities of derivations or word sequences.
- Grammaticality and syntactic structure can be determined for linguistically motivated units other than sentences, e.g. noun phrases. This allows to extract linguistic information even in the face of ungrammatical sentences.

Note that the combination of such a formal grammar with a statistical model is quite different from a statistical

[1] A notable exception is the SuperARV language model by Wang and Harper (2002), which is essentially a class-based language model.

parsing model. A typical statistical parser is completely guided by a statistical model and basically allows for any derivation that is structurally possible. A parser using a formal grammar is restricted by the hard linguistic constraints encoded in the grammar, though these constraints are often selectively relaxed to increase robustness. Thus, the statistical model only needs to "fill the gaps" with quantitative information.

1.3. Formal grammars as language models

Formal grammars have been used as language models since the beginnings of automatic speech recognition (see Erman et al. (1980) for an early example). In spite of that, formal grammars have almost exclusively been applied to natural language understanding tasks in rather restricted domains. Examples of such domains are naval resource management (Price et al., 1988), urban exploration and navigation (Zue et al., 1990), air travel information (Price, 1990), public transport information (Nederhof et al., 1997) and appointment scheduling and travel planning (Wahlster, 2000).

The system by Beutler et al. (2005) is exceptional in that it is exclusively targeted at speech recognition. It was evaluated for an artificial task, namely the transcription of dictation texts for pupils. This task was relatively easy due to the simple language, the small recognizer vocabulary (which completely covered the test utterances) and the good acoustic conditions.

Some studies reported reductions of the word error rate compared to a bigram (Kita and Ward, 1991; Goddeau, 1992; Jurafsky et al., 1995), a trigram (van Noord, 2001) or a 4-gram language model (Beutler et al., 2005). However, the only statistically significant improvements relative to an acknowledged state-of-the-art baseline system were published by Moore et al. (1995) for an air travel information domain. For the Verbmobil system (Wahlster, 2000), effects on the speech recognition performance were not reported.

1.4. Formal grammars for broad domains

There are several reasons why formal grammars are attractive for small-domain speech understanding tasks. First, if parsing is necessary for extracting the semantic interpretation of an utterance, there is no significant overhead in applying the same resources and processing to speech recognition. Further, small domains allow for very restrictive grammars that constrain both the syntax and the semantics of the accepted utterances. In fact, Rayner et al. (2006) created natural language understanding systems by compiling a general grammar into more restricted, domain-specific grammars. For small domains, restrictive grammars can still have a reasonably good coverage, and in applications of human–computer interaction, the user can even be expected to adapt to the grammar with increasing experience. Finally, grammar-based language models are particularly interesting if there is not enough domain-specific data for the training of statistical language models.

It is much more difficult to exploit the strengths of formal grammars in broad-domain large-vocabulary speech recognition. The syntax of broad domains is very productive, and thus many incorrect hypotheses have to be considered grammatical. In our experimental data, the utterance "*arafat räumte fehler seiner regierung ein*" (*arafat admitted mistakes of his government*) gives rise to 37 recognition hypotheses, 25% of which are grammatical. A few examples are shown below:

(1)	*arafat*	*räumte*	*fehler*	*seiner*	*regierung*	*allen*	
	arafat	cleared	mistakes	his	government	to all	
	'arafat cleared mistakes of his government for everyone'						
(2)	*arafat*	*und*	*der*	*fehler*	*seiner*	*regierung*	*allein*
	arafat	and	the	mistake	his	government	only
	'only arafat and the mistake of his government'						
(3)	*arafat*	*wollte*	*fehler*	*seiner*	*regierung*	*einen*	
	arafat	wanted	mistakes	his	government	to unify	
	'arafat wanted to unify mistakes of his government'						

Some of these phrases are clearly marked, but all of them are possible in appropriate contexts and with a semantically plausible choice of words. This example suggests that simply choosing the acoustically best grammatical hypothesis will not work for broad domains. Rather, there should be a component that captures syntactic preferences and allows to compare different grammatical hypotheses.

If information on grammatical correctness has a reduced discriminative power for broad domains, it appears to be even more important to model linguistic constraints as precisely as possible. In Chapter 2 of Sag et al. (2003) it is argued that context-free grammars are not adequate for developing precise grammars with broad coverage. Rather, more sophisticated grammar formalisms are required (see Section 3.1).

However, formal grammars that aim at precision in broad domains are inevitably incomplete. On the one end of the spectrum, there are general grammatical constructions that are notoriously difficult to formalize, for example ellipses, coordinations and parentheses. On the other end,

there is a wealth of idiosyncratic phenomena such as determiner omission in temporal expressions like "*next week*". But even if standard language were completely covered by the grammar, the best available hypotheses may still be ungrammatical due to out-of-vocabulary words, bad acoustic conditions, incorrect utterance boundaries or ungrammatical utterances. This necessitates a robust approach in the sense that syntactic cues should also be exploited for hypotheses that are not accepted by the grammar.

In summary, we argue that a grammar-based approach to broad-domain speech recognition requires a general but precise grammar (and hence a grammar formalism which facilitates the development of such a grammar), a strong model for syntactic preferences and a robust way of integrating these components into speech recognition.

None of the previous approaches meets all of these requirements. Goodine et al. (1991), Goddeau (1992), Jurafsky et al. (1995) and Rayner et al. (2006) used statistical models for syntactic preferences. However, they either considered only grammatical hypotheses or relied on some kind of fallback mechanism to handle ungrammatical hypotheses. On the other hand, Moore et al. (1995), Kiefer et al. (2000), van Noord (2001) and Beutler et al. (2005) achieved robustness by means of partial parsing (see Section 2.3.2) but did not consider syntactic preferences. Finally, only the Verbmobil grammar by Müller and Kasper (2000) and the grammar used in (Beutler et al., 2005) attempted to model general language. These two grammars were based on a linguistically motivated grammar formalism, whereas most other grammars were context-free or even LR(1).

1.5. Outline

In Section 2, we will discuss our approach to integrating formal grammars into speech recognition. Our way of modeling and processing natural language will be presented in Section 3. This section also aims at pointing out some of the difficulties that have to be faced when working with precise models of general language. Section 4 is dedicated to the acquisition of linguistic resources in general and lexical information in particular. Finally, Section 5 reports on our experiments and discusses the results.

2. Integration of syntactic information

2.1. Architecture

The basic architecture is shown in Fig. 1. In a first stage, a speech signal is processed by a baseline speech recognizer and the resulting word lattice is automatically segmented into sub-lattices that roughly represent sentence-like units.

For each sub-lattice, the second stage extracts the N best hypotheses with respect to baseline recognizer score. In our particular case, the baseline recognizer score is a weighted sum of a (logarithmic) acoustic likelihood, an n-gram language model score and a word insertion penalty.

For each hypothesis, a parser determines a unique parse tree and its associated disambiguation score. This score reflects the plausibility of the parse tree: highly plausible trees receive large positive scores and highly implausible trees receive large negative scores.

Finally, a discriminative reranking component chooses the most promising hypothesis for a given sub-lattice. Reranking is based on various features that are extracted from the individual hypotheses, including the baseline recognizer score, the disambiguation score and different properties of the parse tree. These features are used to compute a final score for each hypothesis, and the hypothesis with the maximum score is chosen as the recognition result. Table 1 shows the relevant scores for an example from our experimental data.

Note that the benefit of the syntactic information can easily be assessed by comparing the word error rate of the baseline speech recognizer (i.e. of the first-best hypotheses) to that of the full system.

Also note that the baseline speech recognizer already employs statistical language models, namely n-grams. In the present approach, these models are complemented with a discriminative language model which considers syntactic information. The interaction between these models is twofold. First, the discriminative reranking takes the n-gram language model scores of the speech recognition hypotheses into account. Second, n-gram language models are involved in choosing the N best hypotheses in the first place: the pruning during speech decoding and N-best extraction both rely on n-grams.

The idea of discriminative reranking with syntactic features is not new. A similar setup was used by Collins et al. (2005) and McNeill et al. (2006) to integrate statistical parsers into speech recognition. An actual contribution of this work is our approach of using formal grammars within this framework. This approach includes a way of determining a parse tree and a plausibility score for arbitrary (even ungrammatical) word sequences, as well as novel features for robust parsing and hypothesis reranking.

The remainder of this section will discuss different building blocks of this approach. Section 2.2 provides a short introduction to discriminative reranking with log-linear models. This technique is used in robust parsing (Section 2.3) as well as hypothesis reranking (Section 2.4). Finally, related work is discussed in Section 2.5. Details about the segmentation component can be found in (Kaufmann, 2009).

2.2. Discriminative reranking with log-linear models

Reranking applies to situations where an observation and several competing candidate interpretations are given. The goal of reranking is to find the most likely interpretation for a given observation. Johnson et al. (1999) proposed a reranking approach based on conditional

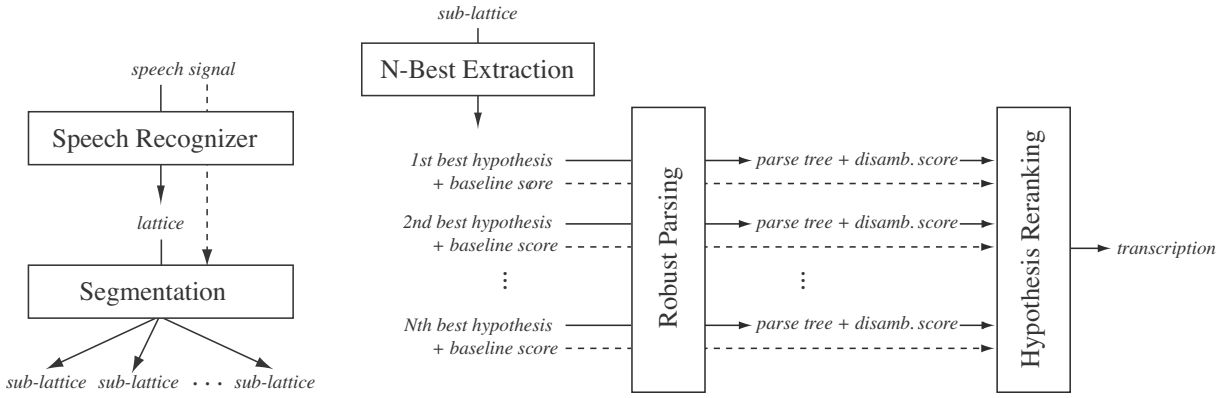

Fig. 1. Basic architecture. An utterance is processed by the baseline speech recognizer and the resulting word lattice is automatically split at potential sentence boundaries. For each sub-lattice, the N best hypotheses are extracted and the most promising hypothesis is chosen based on syntactic information and the scores assigned by the baseline speech recognizer.

Table 1
The five best hypotheses for the utterance "*als Bush im Auto den Reichstag verliess, konnte er sicher sein*" (*when Bush left the Reichstag by car, he could be sure*). Each hypothesis is annotated with the baseline speech recognizer score s_{bas}, the disambiguation score s_{dis} and the final recognition score s_{rec} (the maximum values are printed in bold face). Note that the correct hypothesis on the fifth rank receives the highest final score. The computation of s_{dis} and s_{rec} is explained in Sections 2.3 and 2.4, respectively.

1. *als burschen autor den reichstag verliess konnte er sicher sein*			
	$s_{bas} = \mathbf{-614265.7}$	$s_{dis} = -0.09$	$s_{rec} = -41418.00$
2. *als burschen auto den reichstag verliess konnte er sicher sein*			
	$s_{bas} = -614275.3$	$s_{dis} = -0.11$	$s_{rec} = -41418.50$
3. *als puschen autor den reichstag verliess konnte er sicher sein*			
	$s_{bas} = -614290.1$	$s_{dis} = -1.72$	$s_{rec} = -41420.06$
4. *als wuschen autor den reichstag verliess konnte er sicher sein*			
	$s_{bas} = -614290.5$	$s_{dis} = -2.51$	$s_{rec} = -41420.46$
5. *als bush im auto den reichstag verliess konnte er sicher sein*			
	$s_{bas} = -614291.3$	$s_{dis} = \mathbf{+4.16}$	$s_{rec} = \mathbf{-41416.90}$

log-linear models. In the notation of Charniak and Johnson (2005), the corresponding probability distribution is as follows:

$$P_\theta(y|\mathcal{Y}) = \frac{1}{Z_\theta(\mathcal{Y})} e^{\sum_j \theta_j f_j(y)} \tag{4}$$

$$Z_\theta(\mathcal{Y}) = \sum_{y' \in \mathcal{Y}} e^{\sum_j \theta_j f_j(y')} \tag{5}$$

In the above equations, the observation is implicitly represented by the set $\mathcal{Y}$ of all candidate interpretations that are consistent with the observation. An individual candidate $y \in \mathcal{Y}$ is described by real-valued feature functions $f_j(y)$. An important property of log-linear models is that these feature functions are not assumed to be statistically independent. Given a set of feature weights θ_j, the most likely candidate $y^\star$ can be obtained as follows:

$$y^\star = \operatorname*{argmax}_y P_\theta(y|\mathcal{Y}) = \operatorname*{argmax}_y \sum_j \theta_j f_j(y) \tag{6}$$

Thus, a candidate y maximizes the conditional probability $P_\theta(y|\mathcal{Y})$ if and only if it maximizes the term $\sum_j \theta_j f_j(y)$. This term will subsequently be called the *score* of candidate y.

The model parameters $\theta = (\theta_1, \ldots, \theta_n)$ are chosen to maximize the (regularized) log-likelihood of the training data:

$$L(\theta) = \log \prod_s P_\theta(y^{(s)}|\mathcal{Y}^{(s)}) - \sum_j \frac{\theta_j^2}{2\sigma_j^2} \tag{7}$$

The regularization term $\sum_j \theta_j^2/2\sigma_j^2$ was introduced by Chen and Rosenfeld (1999). It amounts to a Gaussian prior distribution on the feature weights and prevents overfitting by penalizing large weights. We follow Charniak and Johnson (2005) in using a single regularization constant c instead of the different $1/2\sigma_j^2$, which results in the regularization term $c\sum_j \theta_j^2$. The constant c has to be optimized on held-out data. An efficient method for maximizing $L(\theta)$ has been proposed by Malouf (2002), and an implementation was presented in (Charniak and Johnson, 2005).

2.3. Robust parsing

The goal of robust parsing is to determine a unique parse tree for any word sequence. Thus, an approach to robust parsing has to resolve ambiguities by making a good choice among alternative parse trees, and it has to offer a method for dealing with ungrammatical word sequences. These two problems will be discussed in the following. Section 2.3.1 will give an introduction to the predominant approach to disambiguation, which is also adopted in this work. In Section 2.3.2, we describe our approach to robustness. The types of features that we use for robust parsing are outlined in Section 2.3.3.

2.3.1. Parse disambiguation

Parse disambiguation can be regarded as a reranking task where the observation is a word sequence and the candidates are the parse trees that are consistent with this word sequence. A feature function $f_j(y)$ is typically assumed to have to following form:

$$f_j(y) = \sum_{n \in nodes(y)} e_j(n) \tag{8}$$

In the above equation, y denotes a parse tree and $nodes(y)$ is the set of nodes in that parse tree. The indicator function $e_j(n)$ is 1 if a certain "linguistic event" is associated with node n, and 0 otherwise. For example, $e_j(n)$ may be 1 if and only if the node n represents a pronoun. In this case, the feature $f_j(y)$ simply counts the number of pronouns that occur in the parse tree.

The set of parse trees derived by the parser is usually represented as a packed parse forest (Tomita, 1991). A packed parse forest describes an ambiguous structure by means of disjunctive nodes which represent alternative subtrees. Enumerating all unambiguous parse trees that are implicitly represented in a packed parse forest is typically infeasible. However, there exist algorithms for *ambiguity unpacking* that extract the most likely parse trees from a packed parse forest without enumerating all possible trees. Some of these algorithms impose certain restrictions upon the indicator functions e_j.

For this work, we adopted the selective unpacking algorithm by Carrol and Oepen (2005). This algorithm incrementally extracts parse trees in decreasing order of probability. It assumes that the indicator functions $e_j(n)$ only consider subtrees of n with a fixed maximum height. Increasing the maximum height allows for a greater flexibility in feature design but increases the computational cost. We chose the maximum height to be 3, i.e. four levels of nodes could be accessed. An alternative approach would have been the beam search algorithm by Malouf and van Noord (2004). This algorithm allows for arbitrary indicator functions but does not guarantee to find the optimal solution.

Discriminative reranking has emerged as the standard approach to parse disambiguation, in particular for grammars that are not context-free. It has been applied to precise formal grammars by Riezler et al. (2002), Malouf and van Noord (2004) and Toutanova et al. (2005). Collins and Koo (2005) and Charniak and Johnson (2005) used the same technique to improve the output of statistical parsers.

2.3.2. Robustness

Our approach to robustness is based on *partial parsing*. In partial parsing, the parser identifies and analyzes all phrases that can be found somewhere in the given word sequence W. Thus, parsing results in a set of *partial parse trees* spanning different subsequences of the given word sequence.

Robustness can be achieved by allowing to analyze a word sequence as an arbitrary sequence of partial parse trees. Note that such a sequence is guaranteed to exist if a single word is regarded as a degenerate partial parse tree. In the following, a sequence of partial parse trees will be thought of as an *artificial parse tree* that is created by attaching the partial parse trees to a common root node. An example of an artificial parse tree is shown in Fig. 2.

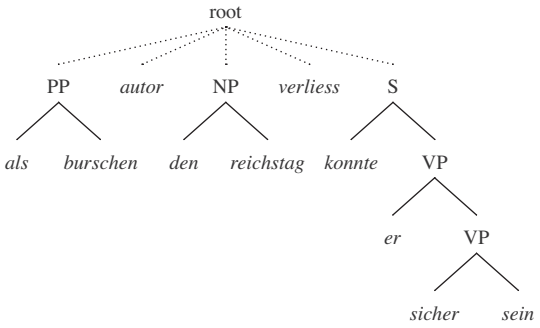

Fig. 2. An artificial parse tree for the (ungrammatical) first hypothesis of Table 1. It consists of five partial parse trees: the prepositional phrase "*als burschen*" (*as busters*), the noun phrase "*den reichstag*" (*the Reichstag*), the sentence "*konnte er sicher sein*" (*could he be sure?*) and the single words *autor* (*author*) and *verliess* (*left*).

The key idea of our approach is to let the disambiguation component determine the most likely *artificial* parse tree (and thus the most likely sequence of partial parse trees). To this end, the reranking model is extended with a set of dedicated feature functions that are sensitive to properties such as the number and types of partial parse trees.

2.3.2.1. Disambiguation with artificial parse trees. Extracting the most likely artificial parse tree from a packed parse forest is straight-forward if the indicator functions $e_j(n)$ can only access subtrees of n and if $e_j(n_{root}) = 0$ for the root node n_{root}. For an artificial parse tree $\bar{y}$ with partial parse trees y_m, it then holds that $score(\bar{y}) = \sum_j \theta_j f_j(\bar{y}) = \sum_m score(y_m)$. Thus, it is sufficient to find a sequence of partial parse trees which maximizes the sum of the individual scores. This can be done by means of dynamic programming. If $L(i,j)$ denotes the maximum score of all partial parse trees spanning the subsequence $(w_i, \ldots, w_j)$ of $W = (w_1, \ldots, w_N)$, the score of the most likely artificial parse tree can be computed as follows:

$$M(l) = \begin{cases} 0 & \text{if } l = 0 \\ \max\limits_{0 \leqslant k < l} \left(M(k) + L(k+1, l)\right) & \text{if } 1 \leqslant l \leqslant N \end{cases}$$

The score of the most likely artificial parse tree is $s_{dis} = M(N)$. In order to recover the actual parse tree, it is necessary to memorize the arguments of all maximizations that are performed when computing L and M.

This algorithm already allows to include certain features that characterize the artificial parse tree. For example, an indicator function $e_j(n)$ can be defined to be 1 if and only if n is the root of a partial parse tree. The respective feature $f_j(y)$ then counts the number of partial parse trees and the feature weight θ_j represents a penalty for introducing an additional partial parse tree. Other features may count partial parse trees with a specific syntactic category, e.g. noun phrases or adverbials. In (Kaufmann, 2009), we showed that this approach can also be extended to less restricted indicator functions, e.g. functions that depend on the actual number of partial parse trees. Such a function could,

for example, indicate that the artificial parse tree contains at least three partial parse trees.

2.3.2.2. Training with artificial parse trees. The training of the discriminative reranking model is complicated by the fact that the number of candidates (i.e. the number of possible artificial parse trees) can be prohibitively large. We adopted an approach that was originally proposed by Osborne (2000) and Malouf and van Noord (2004): we created a manageable candidate set by subsampling the full set of candidates. We did not employ uniform random sampling because this would have led to impoverished sets in which candidates with few partial parse trees are underrepresented. Instead, a training example was constructed by randomly selecting 10 artificial parse trees for each possible number of partial parse trees. The best possible artificial parse tree was added to the candidate set if it was not already part of it.

2.3.2.3. Related work and discussion. Common approaches to robust parsing use simple heuristics to break an ungrammatical word sequence into grammatical parts, completely ignoring the plausibility of partial parse trees (van Noord, 2001; Riezler et al., 2002; Kaplan et al., 2004; Beutler et al., 2005; van Noord et al., 2006). For example, the *fewest-chunks* heuristics minimizes the number of parse trees, even if this means to choose a highly unlikely partial parse tree. Prins and van Noord (2003) address this problem by essentially filtering unlikely parse trees by means of a statistical part-of-speech tagger. However, tagging cannot resolve all ambiguities and it may introduce errors from which the parser cannot recover. In our approach, the quality of a tree sequence naturally includes the quality of the individual trees.

Zhang et al. (2007) used a dedicated model for the probability of breaking a word sequence into a specific sequence of chunks. However, this model is again based on a heuristics, and the authors did not provide an exact and efficient algorithm for finding the optimal sequence of partial parse trees.

2.3.3. Features

This section outlines the most important feature classes that were used for robust parsing. A description of the complete feature set (which consists of roughly 12,000 features) is beyond the scope of this paper and can be found in (Kaufmann, 2009).

2.3.3.1. Partial parse tree features. The partial parse tree features have already been introduced as a key component of our approach to robustness. One feature simply counts the number of partial parse trees. Other features are restricted to single words, certain complete categories (e.g. noun phrases, main clauses or adverbials), or to incomplete categories such as noun phrases with a missing determiner. The latter were included to gather syntactic information in the face of elliptical speech and other out-of-grammar utterances. Other features are 1 if and only if there are at least m partial parse trees. The feature for $m = 2$ is particularly important as it indicates whether there is a single complete parse tree or not.

2.3.3.2. Part-of-speech mismatch features. In many incorrect syntactic analyses, words are assigned the wrong parts of speech. In order to improve disambiguation, we use the output of a statistical part-of-speech tagger. This information is integrated by means of features that count mismatches between the tagger output and the parts of speech that are assumed in the parse tree. Apart from a general mismatch feature, there are features that only consider words with a particular assumed part of speech, e.g. proper noun or pronoun. Prins and van Noord (2003) used tagging information to exclude unlikely lexicon entries from parsing. This stands in contrast to our approach where tagging information can be overruled by other evidence.

2.3.3.3. Generic rule features. The generic rule features aim at covering a large space of linguistic events that are potentially relevant for disambiguation. For each grammar rule, there is a feature that counts how often that rule has been applied in the course of the derivation. More specific features describe the event that a given rule was applied to phrases that belong to specific classes, e.g. that a particular binary rule was applied to a verb and a noun phrase without determiner. Other features count how often a phrase is created by consecutively applying the rules $r_1, r_2, \ldots, r_m$ to a particular element of the phrase, the so-called *head element*. In our experiments, we used $m \in \{2, 3\}$.

2.3.3.4. Miscellaneous features. In addition to these rather general features, there is a variety of very specific features that are motivated by linguistic intuition. These features are used to capture preferences with respect to the constituent order within clauses, the difference in length of two conjuncts and the underspecification of case, just to mention a few.

2.4. Hypothesis reranking

The goal of hypothesis reranking is to choose the most promising speech recognition hypothesis for a given

Table 2
The features used for hypothesis reranking.

Features provided by the baseline speech recognizer
Acoustic log-likelihood
Weighted language model score + word insertion penalty
Baseline recognizer score (the sum of the former two)
Features introducing syntactic information
Disambiguation score of the best artificial parse tree
All features used for disambiguation (see Section 2.3.3)
Features combining syntax and recognizer information
Partial parse tree counts for non-first-best hypotheses
Prosodic partial parse tree features

utterance. This is done by means of a conditional log-linear model that estimates the conditional probability of each hypothesis. In fact, it is sufficient to compare the scores $s_{rec} = \sum_j \theta_j f_j(y)$ of the different hypotheses y, as was pointed out in Section 2.2. The features of the log-linear model are shown in Table 2.

Even though the acoustic log-likelihood and the language model score are already contained in the baseline recognizer score, these two components are represented by dedicated features. This enables the model to change the relative influence of the statistical language model which is now complemented with syntactic information.

As for the syntactic features, it should be noted that the disambiguation model is trained to discriminate between alternative parse trees for the same word sequence. This suggests that the disambiguation score is only partly adequate for comparing parse trees of different hypotheses, even though it turns out to be useful for this purpose (see Section 5.6). For this reason, the disambiguation score is complemented with the full set of features that are used for disambiguation. These features allow to capture syntactic patterns that are characteristic of correct or incorrect hypotheses.

Some of the features combine information about partial parse trees with information that is specific to speech recognition. For example, there is an additional set of features that indicate whether there are at least m partial parse trees. In contrast to the original set, these features are defined to be zero for the first-best hypothesis. They enable the model to selectively penalize non-first-best hypotheses and thus to control the tendency to choose a hypothesis other than the first-best one. The prosodic partial parse tree features were intended to reward partial parse trees that are adjacent to prosodic phrase boundaries. As the benefit of these features was only modest, they are not discussed here.

2.5. Discussion and related work

In the previous sections, we have proposed an approach of integrating formal grammars into automatic speech recognition. The approach offers ways to deal with the limited coverage of precise formal grammars and the fact that incorrect recognition hypotheses are often grammatical. Both problems are tackled by means of partial parsing and considering the plausibility of parse trees.

One novel aspect of this work is the proposed approach to robustness which allows to determine a parse tree and a "plausibility score" for word sequences that are not covered by the grammar. Another novel aspect is our particular way of integrating syntactic information into hypothesis reranking. This includes the use of the disambiguation score and the partial parse tree features. The latter indicate, among others, whether a hypothesis is covered by the grammar or not. This information is already reflected in the disambiguation score, but it can also explicitly be taken into account for hypothesis reranking.

Our approach is based on N-best reranking rather than lattice parsing or even tighter integration into the speech recognition decoder. One reason for this is that parsers for precise broad-coverage grammars are not suited for such processing. They tend to consume too much memory for storing a large number of partial results. Further, efficient search techniques such as A* are not applicable because such parsers typically do not operate from left to right. However, there is evidence that N-best rescoring is sufficient for investigating the benefit of syntactic information. Collins et al. (2004) observed that the best results for speech recognition with statistical parsing models were obtained by means of N-best parsing. Several authors reported that lattice parsing led to gains in processing speed, but to a reduced or only marginally improved accuracy (Roark, 2001a; Hall and Johnson, 2003; Collins et al., 2004).

The most closely related work is that of Collins et al. (2005) and McNeill et al. (2006). They used statistical parsers to derive parse trees for the N best speech recognition hypotheses. The hypotheses were reranked by means of syntactic features that were extracted from the parse trees. As statistical parsers cover arbitrary word sequences by design, robustness was not an issue.

The features of Collins et al. (2005) represented sequences of words and part-of-speech tags, context-free rule applications, head word annotations of non-terminal symbols and dependencies between two head words. Their reported improvements were mostly due to features that are based on part-of-speech tags and did not involve parse trees at all.

McNeill et al. (2006) considered the 20 best parse trees of each hypothesis. This makes the approach less sensitive to errors of the statistical parser and it allows to compute more reliable estimates for the feature values. Their feature set was richer than the one of Collins et al. (2005) and included features for specific syntactic constructions. They further considered the (generative) probability of a parse tree. This probability roughly corresponds to our disambiguation score. However, the latter originates from a discriminatively trained model and thus represents a different kind of information.

3. Formalizing natural language

Although our approach to speech recognition does not rely on a particular grammar formalism, this section will give a very brief introduction to the formalism which was employed in the present work. The reason for this is that the choice of the grammar formalism does have an impact on the practicability of a grammar-based approach. This is particularly true if one attempts to model a large fragment of a natural language by means of restrictive grammar rules.

We argue that with an adequate grammar formalism, the number and complexity of grammar rules is manageable even for precision grammars with a large coverage

of linguistic phenomena. By way of illustration, we will provide a short outline of our German grammar. A more in-depth discussion of both the grammar and the grammar formalism can be found in (Kaufmann, 2009).

3.1. The grammar formalism

In our experiments, we used a Head-driven Phrase Structure Grammar (HPSG). Good introductory textbooks on this formalism are Sag et al. (2003) and – for German HPSG – Müller (2007). The foundations of HPSG were published as Pollard and Sag (1987) and further developed in (Pollard and Sag, 1994).

HPSG is based on context-free phrase structure rules which describe how words or phrases can be combined to form larger linguistic units. But unlike context-free grammars, HPSG represents linguistic units as hierarchical feature structures rather than atomic pre-terminal symbols. Feature structures can describe a linguistic unit on a high level of abstraction. For example, they can specify with which kind of unit some word or phrase can combine and in what way.

A consequence of this high level of abstraction is the *lexicalization of the grammar*: the lexical representation of a word (i.e. its feature structure) bears precise information on how the word interacts with other linguistic units. Thus, a considerable part of the grammar can be thought of as being represented in the lexicon. Consequently, many syntactic peculiarities can be accounted for by simply introducing specific lexicon entries rather than changing the grammar rules.

If linguistic information is adequately organized in feature structures, it is possible to state grammar rules which capture general linguistic concepts. For example, a single rule might account for the way an adjective attaches to a noun or an adverb to a verb. Such a rule would require the respective lexicon entries to specify which kind of linguistic unit they can attach to. The interaction of few general rules can account for a large number of syntactic constructions which otherwise would have to be described individually. However, it is still possible to use construction-specific phrase structure rules where necessary.

In summary, HPSG facilitates the development of large-coverage precision grammars by means of lexicalization and generalization. Lexicalization allows to handle many kinds of exceptions in a simple way, and the ability to capture general linguistic concepts leads to a relatively small number of grammar rules. An example of such a grammar will be outlined in the next section.

3.2. The grammar

Our German grammar is based on diverse literature on German HPSG, most notably Müller (1999), Crysmann (2003, 2005), Müller (2007). Since a detailed description is beyond the scope of this paper, we will only give a broad overview of the grammar. In particular, we will state the number of grammar rules that are dedicated to a specific phenomenon or construction.

Twenty-two grammar rules are used to model the basic linguistic concepts *subcategorization*, *modification*, *extraction* and *extraposition*. Some of these rules are largely language-independent, whereas others are motivated by more language-specific phenomena such as the relatively free word order in German or the so-called *fronting of partial verb phrases*. The formation of the German verbal complex is covered by 3 additional rules. Coordinations (e.g. "*neither the man nor the woman*") are modeled by means of 4 grammar rules. There is one grammar rule for relative clauses and another one for interrogative clauses. Finally, 8 grammar rules are involved in the construction of noun phrases. They include rules for nominalized adjectives ("*der Kleine*", engl. *the little one*), prenominal genitives ("*des Pudels Kern*", engl. *the poodle's core*), post-nominal genitives ("*die Stunde der Wahrheit*", engl. *the moment of truth*) and appositions ("*Präsident Bush*", engl. *president bush*).

The 39 rules that were outlined above represent the grammar for general language use. This grammar is complemented with domain-specific subgrammars which amount to 49 additional rules. These subgrammars describe certain expressions of date and time, as well as forms of address. Further, they account for the fact that the speech recognizer may split certain compound words, e.g. spelled-out numbers ("*ein und zwanzig*"), compound nouns ("*zeitungs bericht*") and acronyms ("*b. m. w.*").

Some expressions follow restricted and very specific variation patterns that are rather cumbersome to handle in HPSG but can easily be expressed by means of regular expressions. An example is "*bis Ende August 2009*" (*until the end of August 2009*). Such expressions are specified in a regular grammar. Each regular expression is mapped to a feature structure which represents the interface to the Head-driven Phrase Structure Grammar.

4. Development of lexical resources

4.1. Introduction

Precise formal grammars require precise lexical information. A noun, for example, can be countable or uncountable, it can denote a title, a personal or geographical name, a quantity or a temporal unit, and it can have four types of sentences as a complement. For verbs, we distinguish about 880 subcategorization frames (i.e. possible sets of complements). Most verbs can be used with several subcategorization frames. For example, the verb *gehen* (*to go*/*walk*) is assigned 16 different frames in our lexicon.

The following sections address the problem of acquiring such lexical information, given a list of word forms that should be covered by the lexicon. In some cases, it is a major problem to identify the lexical units in the first place. Examples are German particle verbs or multiword expressions such as "*nach wie vor*" in German or "*by and large*" in English.

In general, dictionaries or reference books do not directly provide the specific, fine-grained syntactic categories that are required for our purposes. They also tend to represent the prototypical usages of a word and neglect non-standard language use and the numerous variations that can be observed in corpus data. Thus, we decided to rely on data-driven, semi-supervised approaches whenever possible. We did not use treebanks or the output of statistical parsers in order not to weaken the argument that the proposed approach requires relatively little syntactically annotated data. Instead, we mainly relied on a text corpus of 220 million words that was tagged with the TnT part-of-speech tagger by Brants (2000). The corpus consists of news reports from the Frankfurter Rundschau that were published between 2nd January 1997 and 13th April 2002.

The methods that will be described next were used to create a lexicon that covers the 7000 word forms occurring in our experimental data. The lexicon contains about 600 adjectives, 4000 nouns and 3300 verbs. The verbs were annotated with more than 11,000 subcategorization frames. The number of multiword lexemes is about 1200, not including the 1400 acronym entries that were generated automatically. The whole set of lexicon entries gives rise to about 120,000 inflected forms.

Our decision to acquire only the words that occur in the experimental data calls for an explanation: technically, restricting the vocabulary like this gives us information about the test data. We chose to do so because it would have been too laborious to manually acquire the full speech recognizer vocabulary of about 300,000 words. Also, research in (sufficiently effective) methods of automatic lexical acquisition was clearly outside the scope of our work. However, we think that our results are not significantly affected by this decision. First, the acquisition is based on an alphabetically sorted list of about 7000 words, about 30% of which actually appear in the reference transcription. This makes it difficult to guess if a particular word occurs in the transcription and in which particular syntactic contexts. Second, our semi-automatic acquisition process heavily relies on information which is completely unrelated to the test data, for example morphological databases, dictionaries and automatically extracted corpus examples.

4.2. Identification of open-class words

Before determining the syntactic categories, it is necessary to identify the lexical units (*lexemes*) that account for the given set of word forms. This was achieved by sending queries to the canoo.net morphological database. For each word form, the set of matching lexemes was retrieved (with kind permission of Canoo Engineering AG), where a lexeme was represented by its part-of-speech and its inflected forms. The inflectional class and the stem information of each lexeme were automatically inferred from the inflected word forms.

A special treatment was required for particle verbs such as *untergehen* (*to sink*): as the particle (*unter*) and the verb (*gehen*) can occur as two separate words in certain situations, the actual particle verb may not be present in the given set of word forms. In order to check whether a prefix and a verb from the set may form a particle verb, the text corpus was searched for relatively unambiguous non-finite forms such as *unterzugehen* and *untergegangen*.

The identification of personal names and geographical names is difficult for two reasons. First, they are not well covered by dictionaries and linguistic databases. Second, many of them are homonyms of nouns, verbs or adjectives. For each word form, it was manually decided whether it can be used as a given type of proper name (i.e. first name, last name or geographical name). The decision-making was supported by two pieces of information: the output of a classifier that was trained to solve the same task, and the 10 "most informative" corpus examples, i.e. the corpus examples in which the word in question is most likely to be used as a proper name of the given type. Our approach of training such classifiers and determining the most informative examples will be outlined in the next section.

4.3. Acquisition of open-class words

Once all nouns, verbs and adjectives were identified, the next step was to determine their syntactic properties. This was done manually with the help of information that was extracted from the tagged text corpus. For each lexeme, the lexicon developer had to decide whether a given usage (e.g. the uncountable use of a given noun) is possible or not. To help with the decision, the lexicon developer was presented a number of corpus examples in which the given word potentially occurs in that specific usage. This approach is presumably more efficient (and more accurate) than browsing through dictionaries, and the corpus information can be expected to be more reliable than the intuition of the lexicon developer.

A straightforward approach to selecting corpus examples is to specify a regular expression for contexts that are characteristic for the given usage. The symbols of the regular expression can be defined in terms of word forms and/or part-of-speech tags. This approach was found to work reasonably well for certain syntactic properties. For other properties, the value of a corpus example seemed to depend on many different cues whose relative importance is not intuitively clear. For example, the absence of a determiner is not a good indicator for a mass noun, as determiners of count nouns can also be omitted in appropriate contexts such as headlines. The preceding word seems to be more informative. In particular, words like *viel* (*much*) are strong indicators for uncountable use. Our second approach attempts to integrate and weight such cues.

The basic idea is to train a log-linear classifier (i.e. a maximum entropy model) to predict whether a given usage is possible for a given word. Each feature of this classifier essentially searches the corpus and counts how often a

specific cue occurs in the context of the word to be classified. The classifier can be trained by means of an initial lexicon, which results in a set of feature weights. These weights represent the relative importance of their respective cues. Summing up the weights of all cues that are present in a corpus example provides a score which measures the "informativeness" of that example.

This way of acquiring syntactic information turned out to work very well for nouns and adjectives. For verbs, however, the approach was of rather limited use. Most subcategorization information was manually obtained from a dictionary (Duden, 1999). Corpus information was only used for detecting certain types of complements such as prepositional objects or sentential complements.

We did not attempt to acquire lexical information fully automatically, as it was done by Baldwin (2005a), Baldwin (2005b), Nicholson et al. (2008), among others. The main reason is that we wanted the lexical resources to be as accurate as possible. However, it was not investigated how much our approach to speech recognition relies on an accurate lexicon.

4.4. Acquisition of multiword lexemes

Using the terminology of Sag et al. (2002), we are concerned with *lexicalized phrases*, i.e. phrases whose syntactic or semantic properties cannot be derived from their parts. More precisely, we are only interested in syntactic compositionality. For example, the meaning of "*den Löffel abgeben*" (*to kick the bucket*, literally: *to hand in the spoon*) cannot be derived from its parts. From a syntactic point of view, however, this expression is completely regular. This is in contrast to the expression "*Schlange stehen*" (*to queue up*, literally: *to stand snake*). Here, the noun *Schlange* lacks a determiner and the verb *stehen* is used transitively, which both is not possible in general.

We distinguished between 11 types of multiword expressions which reached from completely fixed expressions (e.g. "*ab und zu*", *now and then*) to expressions that undergo internal variation (e.g. "*auf diese/unterschiedliche Weise*", *in this/a different way*) or word order variation (e.g. "*etwas in Kauf nehmen*", *to put up with something*).

Potential multiword expressions were extracted in the standard way: suffix arrays (Manber and Myers, 1990; Yamamoto and Church, 2001) were used to identify all tagged word sequences that occurred at least 5 times in the corpus and that were composed of at most 10 words. For each sequence, it was determined how strongly the word/tag pairs were associated with each other. Pointwise mutual information (Fano, 1961; Church and Hanks, 1990) was chosen as the association measure for word pairs. For sequences of more than two words, the approximation by Schone and Jurafsky (2001) was employed.

The tagged word sequences were sorted in decreasing order of association strength and manually inspected starting from the top-ranked sequence. Each word sequence that was considered to be a relevant multiword expression was marked and later added to the lexicon. The manual inspection was stopped when the event of a relevant multiword expression occurring became too rare. This general procedure was varied by filtering certain word sequences, e.g. sequences that contain words which are not covered by the lexicon, or sequences with a particular part-of-speech pattern.

Our second approach to identifying multiword lexemes exploits the fact that the *prefield* of German main clauses is typically occupied by a single constituent. The prefield generally covers the part to the left of the finite verb. We extracted many instances of the prefield from the text corpus and parsed each of them. The unparsable word sequences were sorted by association strength and inspected manually. Apart from multiword expressions, this also yielded flaws of the grammar and missing syntactic constructions. Examples are postnominal modifiers as in "*er selbst*" (*he himself*) or various expressions of date that were later incorporated in the regular grammar, e.g. "*im September letzten Jahres*" (*in september of last year*). This approach is related to the *error mining* technique which was introduced by van Noord (2004).

5. Experiments and results

5.1. Baseline system and data

Our experiments are based on word lattice output of the 300k LIMSI German broadcast news transcription system which is described in (McTait and Adda-Decker, 2003; Gauvain et al., 2002). The speech recognizer is a multi-pass system that employs continuous density hidden Markov models with Gaussian mixtures for acoustic modeling, 4-gram backoff language models and class-based trigrams. The vocabulary size is 300,000 words.

The lattices cover 6 broadcasts of the German news show *Tagesschau*. The data of three news shows was used in preliminary experiments and for the training of the segmentation component (see Fig. 1). The experiments reported in this work are based on the remaining three news shows (broadcast on 29th April, 15th May and 23rd May 2002). We will subsequently refer to the latter three news shows, unless otherwise noted. These three new shows amount to 603 sentences, 7477 words, or a signal length of 52 minutes. The average sentence length is 12.4 words. The word error rate of the first-best transcriptions is 13.27%. The first-best transcription of one of the word lattices is given below:

```
Doch auf der Internet Seite wird zur besonderen Vorsicht gemahnt <s/> Insbesondere bei Menschen Ansammlungen [silence] {breath} <s/> Auch der Betroffene Reiseveranstalter [silence] zieht Konsequenzen <s/>
```

Besides the tags indicating silence, breath, filler words and potential sentence boundaries, the speech recognition output has certain peculiarities that have to be taken into account by the grammar. In particular, the speech recognizer may split words such as compound nouns (e.g. "*Internet Seite*" instead of "*Internetseite*"), acronyms (e.g. "*B. M. W.*" instead of "*BMW*"), numbers (e.g. "*ein und zwanzig*" instead of "*einundzwanzig*") and prefix verbs (e.g. "*aus handelten*" instead of "*aushandelten*"). This allows the speech recognizer to deal with the potentially infinitely many compound words.

Split compounds are not counted as errors in the evaluation scheme by McTait and Adda-Decker (2003). The evaluation is based on rewrite rules that normalize both the reference transcriptions and the speech recognition hypotheses. Even though syntactic analysis could help to recover the original word forms, we adopted the original evaluation scheme in our experiments.

5.2. Preprocessing

The three news shows were automatically segmented into 609 sentence-like units. From each resulting word lattice, the best hypotheses were extracted incrementally. As many hypotheses only differ in the tags and in the capitalization of words, this information was removed in a normalization step. The extraction of the best hypotheses was stopped as soon as 100 different normalized hypotheses were found.

Lexical acquisition and grammar development were entirely based on the roughly 7000 word forms that occurred in the speech recognition hypotheses, i.e. the language engineer had not other information about the test data. After the development of the linguistic resources, the reference transcriptions of the 609 speech segments were determined and annotated with syntactic structure.

All hypotheses were parsed and the resulting parse forests were stored on disk. For three utterances, some hypotheses could not be parsed exhaustively due to memory limitations (parsing was aborted as soon as more than 5,000,000 feature structure nodes were created). In these cases, we considered only the hypotheses that were scored better than the first unparsable hypothesis. On average, parsing one hypothesis took about 1.5 s with Java 5.0 on a 64 bit Linux work station with AMD Opteron 2218 CPUs and 8 GB RAM.

The part-of-speech tagger for extracting the part-of-speech features was trained on a transformed version of the TIGER corpus (Brants et al., 2002). The tagger and the training data are described in (Kaufmann, 2009).

5.3. Experimental setup

In order to make the best use of the available test data, we followed a 10-fold cross-validation scheme, i.e. the hypothesis reranking model was evaluated on each fold after being trained on the 9 remaining folds. Note that the training of the hypothesis reranking model relies on the output of the disambiguation model, namely the most likely parse trees and the respective disambiguation scores. This output is produced by means of an embedded cross-validation cycle: the disambiguation model produces output for one fold after being trained on the remaining 8 folds (the fold for evaluating the hypothesis reranking model has to be excluded from the training).

The utterances were randomly assigned to the 10 folds. Cross-validation with random assignment is problematic for tasks in which similar observations tend to occur in close proximity: the similar observations are likely to be assigned to different folds and thus can be learned by the model. This would in fact be a problem if our models would consider actual word forms (which tend to be clustered within segments of the same topic). However, as the models are based on syntactic categories rather than word forms, such clustering effects can be regarded as negligible.

The training data for the disambiguation model was complemented with 447 syntactically annotated broadcast news transcripts from earlier experiments. There is a slight mismatch between this additional training data and the data from our experiments: the transcripts have correct sentence boundaries and represent only the sections with low word error rates. Overall, the training data for the disambiguation model amounts to about 990 sentences.

The conditional log-linear models were trained with the reranking software by Charniak and Johnson (2005). The regularization parameter c was set to 13 for disambiguation and 30 for hypothesis reranking. These values were found to work well in earlier experiments with similar data and feature sets.

5.4. Broadcast news transcription results

Table 3 shows that automatic segmentation and 100-best reranking reduced the word error rate by 9.7% relative. This improvement is statistically significant at a level of less than 0.1% according to both the Matched Pairs Sentence-Segment Word Error test (MAPSSWE) and the McNemar test on sentence level (Gillick and Cox, 1989).

It is sometimes suspected that syntactic information mainly helps in getting inflectional endings right, particularly for highly inflected languages such as German. In order to check this assumption, we compared the output of the baseline system to the output of the reranking component and counted how often an error was corrected by

Table 3
The word error rate after reranking with syntactic information compared to the first-best word error rate of the baseline system and the 100-best oracle error rate.

Approach	Word error rate
Baseline system	13.27%
+ Syntactic information	11.98% (**−9.7% relative**)
100-Best oracle	6.32%

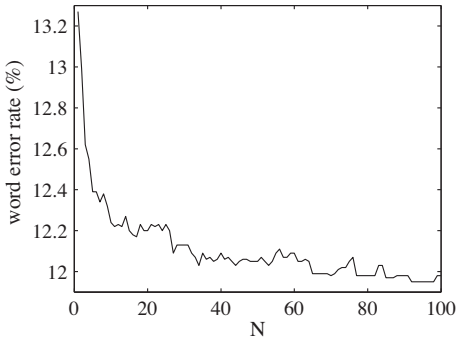

Fig. 3. The word error rates that result from reranking the N best speech recognition hypotheses, $1 \leqslant N \leqslant 100$.

hypothesizing a different inflectional ending, discounting the errors that were introduced due to the same reason. We found that only 27% of the net error corrections involved changing inflectional endings. The remaining corrections were due to introducing a stem or a function word that was not present in the first-best hypothesis.

Another common notion is that formal grammars improve speech recognition essentially by choosing a grammatical hypothesis and that they fail to identify good hypotheses if these hypotheses are ungrammatical. Again, our experimental data suggests that this is not the case. We counted the net error corrections of those utterances in which hypothesis reranking chose a hypothesis with two or more partial parse trees. It was observed that these utterances account for about 37% of the net error corrections.

In order to assess the impact of the number of hypotheses N on the word error rate, we repeated the experiment for all N in 1..100. We did not try higher values of N as the lexicon is only guaranteed to cover the words of the 100 best hypotheses. The results are shown in Fig. 3. The 10 best hypotheses alone account for an improvement of about 1% absolute (7.8% relative). For higher values of N, the word error rate starts to level off. However, there still seems to be room for further improvement beyond $N = 100$.

5.5. The influence of the recognition task

In order to measure the influence of different characteristics of the recognition task, we performed experiments for variants of the original task. The first experiment was identical to the original one with the difference that the sentence boundaries were determined manually. This can be thought of as assuming a perfect sentence boundary detection, or as approaching a dictation scenario where sentence boundaries are clearly marked. As shown in Table 4, manual segmentation led a to relative improvement of 12.2% over the baseline. The contribution of manual segmentation was only weakly significant (4.8% for the MAPSSWE test). However, our results are in line with McNeill et al. (2006), who observed that speech recognition with syntactic features benefited from manual segmentation.

Table 4
The relative improvements for two artificially simplified speech recognition tasks. The first experiment assumes correct segmentation and is based on manually determined sentence boundaries. The second experiment is performed on a subset of the data (anchor speaker, correspondent and weather report sections).

Approach	Word error rate
Baseline	13.27%
+ Syntactic information	11.98% (−9.7% relative)
+ Correct segmentation	11.67% (**−12.2% relative**)
Baseline (selected sections)	10.91%
+ Syntactic information	9.41% (**−13.7% relative**)

The goal of the second experiment was to assess the benefit of syntactic information for a lower baseline word error rate. This was achieved by restricting the data to anchor speaker sections, correspondent speech and weather reports, which amounts to a total of 506 utterances. The sentence boundaries were determined automatically. Table 4 shows that the lower baseline word error rate led to a larger relative improvement. This confirms the intuition that a language model which makes extensive use of non-local information is particularly good at correcting errors within contexts that are largely correct. Beutler (2007) made a similar observation when varying the out-of-vocabulary rate of his baseline speech recognizer.

In order to get an idea of how much improvement can be expected for spontaneous speech, we performed an experiment on the 92 segments that appear to have been uttered spontaneously. It was found that syntactic information reduced the word error rate by 5.6% relative, compared to a baseline of 22.03%. However, these results are by no means conclusive. First, the set of utterances is very small, both for training and for evaluation. Second, the situations with spontaneous speech are often difficult for the baseline speech recognizer because of background noise or insufficient speaker adaptation. The improvements are likely to be larger under more controlled conditions.

5.6. The influence of syntactic information

In this section, it is investigated how the choice of features and the amount of syntactic information influence the improvement relative to the baseline system. The results are shown in Table 5.

Model 1 involved no statistical information on syntactic preferences. The artificial parse trees were extracted by

Table 5
Word error rates for different sets of features.

Model		Word error rate
1	No statistics	12.70% (−4.3% rel.)
2	Score features	12.24% (−7.7% rel.)
3	+ Partial parse trees	12.05% (−9.2% rel.)
	Full model	11.98% (−9.7% rel.)
4	No disambiguation score	12.39% (−6.6% rel.)

means of the fewest-chunks heuristics, and the feature set for hypothesis reranking was restricted to the baseline recognizer scores and the features based on partial parse tree counts. It can be seen that the lack of statistical information substantially reduces the benefit of our approach. This reduction is statistically significant on a level of <0.1% according to the McNemar test and the MAPSSWE test.

Model 2 employs the complete feature set for disambiguation, but lets hypothesis reranking only consider the disambiguation score and the baseline recognizer scores. This model significantly improves on model 1 (<1.4% according to the MAPSSWE test and <0.1% according to the McNemar test). Model 3 additionally includes the complete set of partial parse tree features in hypothesis reranking (see Sections 2.3.3 and 2.4). The word error rate of this model is already very close to that of the full model.

It is remarkable that model 2 achieves a relatively large improvement even though its sole source of syntactic information is the disambiguation score. The importance of this feature is confirmed by the fact that removing the disambiguation score from the full model (model 4) significantly increases the word error rate (<0.3% according to the MAPSSWE test). Note that in model 4, hypothesis reranking employs all features that are used to compute the disambiguation score. However, the respective feature weights are trained to discriminate between good and bad hypotheses. This seems to be a less effective way of learning syntactic preferences.

It was further investigated to which extent the improvement depends on the precision of the grammar. The original grammar was modified by disabling all rules that participate in the formation of verbal complexes, verb phrases and adjective phrases. The resulting grammar still enforced the agreement between determiners, adjectives and nouns, as well as case agreement between prepositions and their complement noun phrases. The full model in combination with this chunk grammar achieved a word error rate of 12.78%, which corresponds to a relative reduction of 3.7%. This reduction is still significant (0.4% for the MAPSSWE test and 0.1% for the McNemar test).

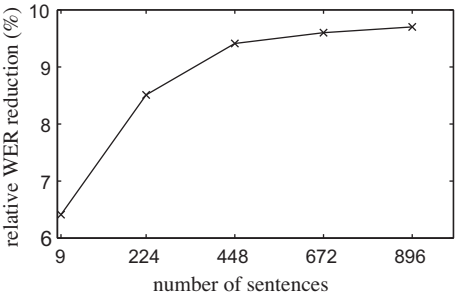

Fig. 4. The effect of the amount of syntactically annotated training data on the relative reduction of the word error rate. Note that the set of 896 sentences represents the full amount of training data. This set consists of the 447 sentences from earlier experiments and 90% of the 609 current sentences (only 90% because of the cross-validation scheme).

We also varied the amount of syntactically annotated data for the training of the disambiguation model by means of random subsampling. The influence of this parameter on the word error rate is shown in Fig. 4. It can be seen that the number of training sentences can be halved without significantly impairing the speech recognition performance. This confirms our claim that a relatively small amount of syntactically annotated data is sufficient for the training of our model. Note that the hypothesis reranking model also employs syntactic features and thus can to some extent compensate for the lack of training data.

Finally, we attempted to evaluate the contribution of our approach to robust parsing. To this end, we performed an experiment in which the disambiguation component chose the partial parse trees according to the *fewest-chunks method*. Note that the fewest-chunks method essentially splits a sentence into segments, each of which corresponds to a partial parse tree. For a given segment, we still choose the most likely parse tree according to our model. We observed a relative improvement of 9.01%, which is only slightly (and not significantly) lower than the improvement with the original approach (9.70%).

However, it must be noted that even for the fewest-chunks method, we still consider the selected partial parse trees as an artificial parse tree and apply our disambiguation model to that tree. Thus, the main contribution of our robustness approach is not the (possibly) better choice of chunks, but the fact that it produces a disambiguation score even for ungrammatical hypotheses. As we have seen in Table 5, this score contributes significantly to the overall improvement.

5.7. Coverage of grammar and lexicon

In order to assess the coverage of the grammar and the lexicon, the reference transcriptions of the manually determined sentence segments were parsed. As the words occurring in the reference transcriptions are not necessarily covered by the lexicon, missing words were included beforehand. For each sentence, the parse tree that best matched the reference syntax tree was extracted and examined. A correct complete parse tree was found for 61% of the sentences. Another 7.3% of the sentences were ellipses for which all parts could be correctly analyzed by the parser. Only 3% of the sentences were considered to be clearly ungrammatical in the sense of incorrect language use.

It was also attempted to determine the different types of errors that prevented sentences from being analyzed correctly. It was found that lexicon errors account for about 30% of the total number of errors. More than half of the lexicon errors were due to missing verb valence information. Another common type of lexicon error were missing multiword lexemes such as "*kirch media*", as well as lexicalized phrases like "*aus dem stand heraus*" (*straight away*). The remaining lexicon errors were due to incorrect or missing single-word lexicon entries, mostly for function words. None of the lexicon-related errors involved an incorrect

noun entry. This confirms our impression that our method for acquiring nouns works quite well. On the other hand, the acquisition of verb valence information appears to be a major problem which might benefit from additional data-driven techniques.

Only 5% of the errors could be attributed to cases where the grammar failed to account for a construction that was supposed to be covered by the grammar. For example, the phrase "*das depot, das umgeben von wohngebieten ist*" (*the depot which is surrounded by residential areas*) cannot be analyzed because the grammar prohibits the interruption of verbal complexes (in this example *umgeben ist*) by complements.

About 41% of the errors were due to general constructions that are not covered by the grammar. Some of these constructions could be incorporated into the grammar with moderate effort, e.g. certain types of appositions, free relative clauses and sentential relative clauses. Other constructions are notoriously difficult to formalize or would give rise to a large amount of ambiguity. These constructions include coordinations (asyndetic coordinations, asymmetric coordinations and coordinations of constituents with different categories), parentheses and ellipses. Another important phenomenon of this type is the omission of mandatory determiners such as *auslöser* (*trigger*) in "*auslöser war eine herabstufung des unternehmens*" (*the downgrading of the company was the trigger*).

On the other hand, idiosyncratic or very rare constructions accounted for about 24% of the errors. An example is the construction "*X für X*" (*X by X*) as in the utterance "*in hebron durchkämmten die soldaten haus für haus*" (*in hebron the soldiers combed house by house*). The main problem with such constructions is that there are many of them and they each require a dedicated grammar rule or a special lexicon entry. In fact, it is not trivial to discover such constructions in the first place.

6. Conclusions

6.1. Discussion

In this paper, we have described and evaluated an approach to integrating a precise formal grammar into statistical speech recognition. Even though formal grammars represent natural language in the form of hard linguistic constraints, these constraints are applied in a "soft" way: syntactic information is extracted from arbitrary hypotheses (including hypotheses that are not covered by the grammar) and weighted by means of statistical models. The approach was implemented and applied to a German broadcast news transcription task. Our experiments confirmed that a precise formal grammar can significantly improve large-vocabulary speech recognition, even for a broad domain and a competitive baseline system.

The automatic transcription of broadcast news is subject to out-of-vocabulary words, unknown sentence boundaries and – to some extent – bad acoustic conditions. Thus, the chosen task was by no means simple. Nevertheless, syntactic information may be particularly beneficial for broadcast news transcription as this task involves mostly prepared, grammatical speech. We do not provide a conclusive evaluation on spontaneous speech. Yet we believe that our approach would still yield improvements on such tasks: even ungrammatical spontaneous utterances are likely to contain phrases that provide useful syntactic information. It might also be possible to adapt to the nature of spontaneous speech by extending the grammar or by using preprocessing components that deal with self-repairs and related phenomena (see Moore et al. (1995), for an example).

We have investigated different properties of our approach. In particular, it was observed that grammaticality information alone led to a relatively small (though still statistically significant) improvement. However, the improvement was more than doubled by adding statistical information, even if the statistical models were trained with a very small amount of data (about 450 sentences or 5200 words of syntactically annotated text and 550 N-best lists). This effect may be explained by the precision of the grammar and the lexicon: the grammar still accepts many implausible sentences, but in order to do so, it has to resort to certain constructions that can easily be penalized by the statistical models. Examples of such constructions are prenominal genitives and nominalized adjective phrases, which indeed are strongly penalized by the log-linear models. These results suggest that the qualitative information expressed in the grammar and the lexicon greatly reduces the need for quantitative information, i.e. treebank data.

The small amount of treebank data is a distinctive characteristic of the proposed approach in comparison to the related work of Collins et al. (2005) and McNeill et al. (2006). These approaches are based on statistical parsers that were trained on about one million words of syntactically annotated text. It is difficult to compare the effort of creating a treebank to that of writing a precision grammar and acquiring a large lexicon. Both tasks are very laborious and require expert knowledge. However, it is likely that qualitative information can be transferred to new domains more easily.

Another characteristics is that our approach does not consider head word information at all. Thus, semantic aspects like common combinations of a verb and the head noun of its subject are not captured. This is again in contrast to the work of Collins et al. (2005) and McNeill et al. (2006), who made extensive use of this sort of information. We conclude that the proposed approach is, to some degree, orthogonal to statistical methods that consider head-to-head dependencies. How large the actual gain from combining such statistical methods with rule-based approaches would be and how head word information could be incorporated into our approach are open questions.

6.2. Outlook

The primary objective of this work was to investigate how formal grammars can be used to improve large-vocabulary continuous speech recognition, and how much improvement can be expected from this kind of information.

Although our experiments are valid, it is unclear whether the proposed approach could be used in an operative speech recognition system. The problem is twofold. First, the parsing of the recognition hypotheses would have to be much more efficient. Second, the manual creation of a precise lexicon would be infeasible for a speech recognizer vocabulary with hundreds of thousands of word forms.

One solution could be to improve the current methods, e.g. to develop a more efficient parser and a sufficiently accurate technique for automatic lexical acquisition. Another solution could be to use a less precise grammar and complement it with a stronger model for syntactic preferences. For example, the lexicon might specify only those syntactic properties that can reliably be extracted from text corpora, whereas the other properties remain underspecified. The statistical models could represent preferences for these properties, possibly falling back on less reliable cues present in the text corpora.

Even though we have placed a strong emphasis on precise linguistic knowledge, we are aware that this need not be the optimal way for exploiting syntactic information in language modeling. It thus seems reasonable to explore further the continuum between fully statistical language models and precise formal grammars.

Acknowledgements

This work was supported by the Swiss National Science Foundation. We cordially thank Jean-Luc Gauvain for providing us with word lattices from the LIMSI German broadcast news transcription system. We further thank Canoo Engineering AG for granting us access to their morphological database.

References

Baldwin, T., 2005a. Bootstrapping deep lexical resources: resources for courses. In: Proc. ACL-SIGLEX Workshop on Deep Lexical Acquisition. Ann Arbor, USA, pp. 67–76.

Baldwin, T., 2005b. General-purpose lexical acquisition: Procedures, questions and results. In: Proceedings of the 6th Meeting of the Pacific Association for Computational Linguistics. pp. 23–32.

Beutler, R., 2007. Improving speech recognition through linguistic knowledge. Ph.D. thesis, ETH Zurich.

Beutler, R., Kaufmann, T., Pfister, B., 2005. Integrating a non-probabilistic grammar into large vocabulary continuous speech recognition. In: Proc. IEEE ASRU Workshop. San Juan, Puerto Rico, pp. 104–109.

Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G., 2002. The TIGER treebank. In: Proc. Workshop on Treebanks and Linguistic Theories. Sozopol, Bulgaria.

Brants, T., 2000. TnT: A statistical part-of-speech tagger. In: Proceedings of the 6th Conference on Applied Natural Language Processing. Seattle, Washington, pp. 224–231.

Carrol, J., Oepen, S., 2005. High efficiency realization for a wide-coverage unification grammar. In: Proceedings of the International Joint Conference on Natural Language Processing. Jeju Island, Corea, pp. 165 – 176.

Charniak, E., 2001. Immediate-head parsing for language models. In: Proc. ACL'01. Toulouse, France, pp. 124–131.

Charniak, E., Johnson, M., 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. Proc. ACL'05, 173–180.

Chelba, C., Jelinek, F., 2000. Structured language modeling. Comput. Speech Lang. 14 (4), 283–332.

Chen, S.F., Rosenfeld, R., 1999. A Gaussian prior for smoothing maximum entropy models. Tech. rep. Carnegie Mellon University, Pittsburgh, PA.

Church, K.W., Hanks, P., 1990. Word association norms, mutual information, and lexicography. Comput. Linguist. 16 (1), 22–29.

Collins, C., Carpenter, B., Penn, G., 2004. Head-driven parsing for word lattices. In: Proc. ACL.

Collins, M., Koo, T., 2005. Discriminative reranking for natural language parsing. Comput. Linguist. 31 (1), 25–69.

Collins, M., Roark, B., Saraclar, M., 2005. Discriminative syntactic language modeling for speech recognition. In: Proc. ACL'05. Ann Arbor, Michigan, pp. 507–514.

Crysmann, B., 2003. On the efficient implementation of German verb placement in HPSG. In: Proc. RANLP.

Crysmann, B., 2005. Relative clause extraposition in German: An efficient and portable implementation. Res. Lang. Comput. 3 (1), 61–82.

Duden, 1999. Das grosse Wörterbuch der deutschen Sprache in 10 Bänden, 3. Auflage. Dudenverlag, Mannheim, Leipzig, Wien, Zürich.

Erman, L.D., Hayes-Roth, F., Lesser, V.R., Reddy, D.R., 1980. The Hearsay-II speech-understanding system: integrating knowledge to resolve uncertainty. ACM Comput. Surv. 12 (2), 213–253.

Fano, R.M., 1961. Transmission of Information. MIT press.

Gauvain, J.L., Lamel, L., Adda, G., 2002. The LIMSI broadcast news transcription system. Speech Comm. 37 (1–2), 89–108.

Gillick, L., Cox, S., 1989. Some statistical issues in the comparison of speech recognition algorithms. In: Proc. ICASSP'89. pp. 532–535.

Goddeau, D., 1992. Using probabilistic shift-reduce parsing in speech recognition systems. In: Proc. ICSLP.

Goodine, D., Seneff, S., Hirschman, L., Phillips, M., 1991. Full integration of speech and language understanding in the MIT spoken language system. In: Proc. 2nd European Conf. on Speech Communication and Technology.

Hall, K., Johnson, M., 2003. Language modeling using efficient best-first bottom-up parsing. In: Proc. IEEE Automatic Speech Recognition and Understanding Workshop. pp. 507–512.

Johnson, M., Geman, S., Canon, S., Chi, Z., Riezler, S., 1999. Estimators for stochastic "unification-based" grammars. In: Proc. ACL'99. College Park, Maryland, pp. 535–541.

Jurafsky, D., Wooters, C., Segal, J., Stolcke, A., Fosler, E., Tajchman, G., Morgan, N., 1995. Using a stochastic context-free grammar as a language model for speech recognition. In: Proc. ICASSP'95, Vol. 1. pp. 189–189.

Kaplan, R.M., Riezler, S., King, T.H., Maxwell III, J.T., Vasserman, A., Crouch, R.S., 2004. Speed and accuracy in shallow and deep stochastic parsing. In: Proc. HLT-NAACL'04. pp. 97–104.

Kaufmann, T., 2009. A rule-based language model for speech recognition. Ph.D. Thesis, ETH Zürich.

Kaufmann, T., Ewender, T., Pfister, B., 2009. Improving broadcast news transcription with a precision grammar and discriminative reranking. In: Proc. Interspeech'09. Brighton, England.

Kiefer, B., Krieger, H.-U., Nederhof, M.-J., 2000. Efficient and robust parsing of word hypotheses graphs. In: Wahlster, W. (Ed.), Verbmobil. In: Foundations of Speech-to-Speech Translation. Springer, Berlin, pp. 280–295.

Kita, K., Ward, W.H., 1991. Incorporating LR parsing into SPHINX. In: Proc. ICASSP'91, pp. 269–272.

Malouf, R., 2002. A comparison of algorithms for maximum entropy parameter estimation. In: Proc. Internat. Conf. On Computational Linguistics, pp. 1–7.

Malouf, R., van Noord, G., 2004. Wide coverage parsing with stochastic attribute value grammars. In: Proc. IJCNLP-04 Workshop: Beyond Shallow Analyses – Formalisms and Statistical Modeling for Deep Analyses.

Manber, U., Myers, G., 1990. Suffix arrays: a new method for on-line string searches. In: Proc. 1st Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 319–327.

McNeill, W.P., Kahn, J.G., Hillard, D.L., Ostendorf, M., 2006. Parse structure and segmentation for improving speech recognition. In: Proc. IEEE Spoken Language Technology Workshop, pp. 90–93.

McTait, K., Adda-Decker, M., 2003. The 300k LIMSI German broadcast news transcription system. In: Proc. Eurospeech'03. Geneva, Switzerland.

Moore, R., Appelt, D., Dowding, J., Gawron, J.M., Moran, D., 1995. Combining linguistic and statistical knowledge sources in natural-language processing for ATIS. In: Proc. ARPA Spoken Language Systems Technology Workshop.

Müller, S., 1999. Deutsche Syntax deklarativ. Head-Driven Phrase Structure Grammar für das Deutsche. No. 394 in Linguistische Arbeiten. Max Niemeyer Verlag, Tübingen.

Müller, S., 2007. Head-Driven Phrase Structure Grammar: Eine Einführung. Stauffenburg Einführungen, Nr. 17. Stauffenburg Verlag, Tübingen.

Müller, S., Kasper, W., 2000. HPSG analysis of German. In: Wahlster, W. (Ed.), Verbmobil: Foundations of Speech-to-Speech Translation. In: Artificial Intelligence. Springer-Verlag, Berlin, Heidelberg, New York, pp. 238–253.

Nederhof, M.J., Bouma, G., Koeling, R., van Noord, G., 1997. Grammatical analysis in the OVIS spoken-dialogue system. In: Proc. ACL/EACL Workshop on Spoken Dialog Systems, pp. 66–73.

Nicholson, J., Kordoni, V., Zhang, Y., Baldwin, T., Dridan, R., 2008. Evaluating and extending the coverage of HPSG grammars. In: Proc. 6th International Conference on Language Resources and Evaluation. Marrakech, Morocco.

Osborne, M., 2000. Estimation of stochastic attribute-value grammars using an informative sample. In: Proc. 18th Internat. Conf. on Computational Linguistics, pp. 586–592.

Pollard, C.J., Sag, I.A., 1987. Information-based syntax and semantics. In: CSLI Lecture Notes, Vol. 1. CSLI Publications, Stanford University, No. 13.

Pollard, C.J., Sag, I.A., 1994. Head-driven Phrase Structure Grammar. University of Chicago Press.

Price, P., 1990. Evaluation of spoken language systems: the ATIS domain. In: Proceedings of the Workshop on Speech and Natural Language, pp. 91–95.

Price, P., Fisher, W.M., Bernstein, J., Pallett, D.S., 1988. The DARPA 1000-word resource management database for continuous speech recognition. In: Proc. ICASSP'88, Vol. 1, pp. 651–654.

Prins, R., van Noord, G., 2003. Reinforcing parser preferences through tagging. Traitement Automatique des Langues 44 (3), 121–139.

Rayner, M., Hockey, B.A., Bouillon, P., et al., 2006. Putting Linguistics into Speech Recognition: The Regulus Grammar Compiler. MIT Press.

Riezler, S., King, T.H., Kaplan, R.M., Crouch, R., Maxwell III, J.T., Johnson, M., 2002. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. Proc. ACL'02, pp. 271–278.

Roark, B., 2001a. Markov parsing: Lattice rescoring with a statistical parser. In: Proc. ACL'01. Philadelphia, USA, pp. 287–294.

Roark, B., 2001b. Probabilistic top-down parsing and language modeling. Comput. Linguist. 27 (2), 249–276.

Sag, I.A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D., 2002. Multiword expressions: a pain in the neck for NLP. Lect. Notes Comput. Sci., 1–15.

Sag, I.A., Wasow, T., Bender, E.M., 2003. Syntactic Theory: A Formal Introduction, second ed. CSLI Publications, Stanford.

Schone, P., Jurafsky, D., 2001. Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In: Proc. 6th Conf. on Empirical Methods in Natural Language Processing, pp. 100–108.

Tomita, M., 1991. Generalized LR Parsing. Springer.

Toutanova, K., Manning, C.D., Flickinger, D., Oepen, S., 2005. Stochastic HPSG parse disambiguation using the Redwoods Corpus. Res. Lang. Comput. 3 (1), 83–105.

van Noord, G., 2001. Robust parsing of word graphs. In: Junqua, J.-C., van Noord, G. (Eds.), Robustness in Language and Speech Technology. Kluwer Academic Publishers, Dordrecht.

van Noord, G., 2004. Error mining for wide-coverage grammar engineering. In: Proc. ACL.

van Noord, G., et al., 2006. At last parsing is now operational. In: Actes de la 13e conférence sur le traitement automatique des langues naturelles, pp. 20–42.

Wahlster, W. (Ed.), 2000. Verbmobil: Foundations of Speech-to-Speech Translation. Springer, Berlin.

Wang, W., Harper, M.P., 2002. The SuperARV language model: investigating the effectiveness of tightly integrating multiple knowledge sources. In: Proc. Conf. on Empirical Methods in Natural Language Processing, Vol. 10, pp. 238–247.

Wang, W., Harper, M.P., 2003. Language modeling using a statistical dependency grammar parser. In: Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 519–524.

Wang, W., Stolcke, A., Harper, M.P., 2004. The use of a linguistically motivated language model in conversational speech recognition. In: Proc. ICASSP'04, pp. 261–264.

Xu, P., Chelba, C., Jelinek, F., 2002. A study on richer syntactic dependencies for structured language modeling. In: Proc. ACL'02. Philadelphia, USA, pp. 191–198.

Yamamoto, M., Church, K.W., 2001. Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. Comput. Linguist. 27 (1), 1–30.

Zhang, Y., Kordoni, V., Fitzgerald, E., 2007. Partial parse selection for robust deep processing. In: Proc. Workshop on Deep Linguistic Processing, pp. 128–135.

Zue, V., Glass, J., Goodine, D., Leung, H., Phillips, M., Polifroni, J., Seneff, S., 1990. The VOYAGER speech understanding system: preliminary development and evaluation. In: Proc. ICASSP'90, pp. 73–76.