Pseudo-Syntactic Language Modeling for Disfluent Speech Recognition

Michael McGreevy

mcgreevy@idiap.ch

Speech and Audio Research Laboratory
School of EESE, QUT
GPO Box 2434, QLD, Australia

IDIAP Research Institute
CP 592, rue du Simplon 4,
1920 Martigny, Switzerland

Abstract

Language models for speech recognition are generally trained on text corpora. Since these corpora do not contain the disfluencies found in natural speech, there is a train/test mismatch when these models are applied to conversational speech. In this work we investigate a language model (LM) designed to model these disfluencies as a syntactic process. By modeling self-corrections we obtain an improvement over our baseline syntactic model. We also obtain a 30% relative reduction in perplexity from the best performing standard N-gram model when we interpolate it with our syntactically derived models.

1. Introduction

Speech between humans is usually an *interactive* exercise due to feedback, both visual and verbal, from the listener(s). This interactive property manifests itself in a variety of ways:

1. speech can be cut off mid-utterance, due to interjections, external events, etc.
2. speakers make more corrections to what they say, and
3. speech contains other disfluencies, such as hesitations.

We will focus on the second item.

Corrections in speech make the task of language modeling more difficult. Since language models are generally trained on text, and text does not contain disfluencies, we are faced with a mismatch between the training data and the testing data. If we are able to model the correction process, it may be possible to adapt a model trained on text for the modeling of interactive speech.

We propose that corrections in speech are a syntactic process, which is well supported by the fact that the way a sentence is parsed determines (at least in part) its meaning. This is demonstrated by the two different parses for the sentence in Figure 1 that result from applying the grammatical productions in Table 1. The fact that the listener is usually able to infer the correct meaning from a corrected sentence implies that the correction process involves performing an operation on the parse tree. For this reason, a number of researchers have argued that corrections are indeed a syntactic process, and have proposed rules which govern this process (Carbonell and Hayes 1983; Hindle 1983; Levelt 1989; McKelvie 1998). This is our motivation for developing syntactic language models of conversational speech. Furthermore, according to Schegloff, the correction process is reasonably consistent across languages (Schegloff 1987); this contrasts with the state of the art N-gram approach, which is primarily suited to languages in which "word order is important and the strongest contextual effects tend to come from near neighbours [such as English]" (Young 1996).

In this work we investigate two similar yet distinct types of syntactic language model. One contains production rules which aim to model a specific type of self correction, while the other does not. We find that modeling corrections syntactically leads to a reduction in perplexity compared to a syntactic model that does not take corrections into account.

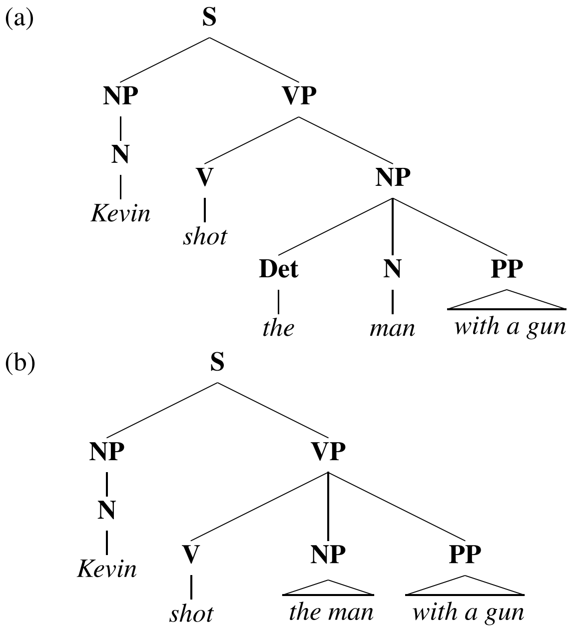

Figure 1: *A sentence parsed in two different ways with two corresponding meanings: (a) "Kevin shot the man who had a gun." (b) "Kevin used a gun to shoot the man."*

2. A tale of two models

The role of a language model in an automatic speech recognition (ASR) system is to calculate the quantity

$$P(w_1^n) \tag{1}$$

Table 1: *A simple grammar.*

Constituent	Abbreviation	Productions
Sentence	**S**	**S** → **NP VP**
Noun Phrase	**NP**	**NP** → **N** **NP** → **Det N** **NP** → **Det N PP**
Verb Phrase	**VP**	**VP** → **V NP** **VP** → **V NP PP**
Preposition Phrase	**PP**	**PP** → **P NP**
Noun	**N**	**N** → Kevin **N** → man **N** → gun
Verb	**V**	**V** → shot
Determinative	**Det**	**Det** → the **Det** → a
Preposition	**P**	**P** → with

where w_j^k refers to the sequence of words $w_j..w_k$ and n is the number of words in a hypothesized utterance.

There are a number of ways to estimate this distribution; we will consider two of them – the N-gram and the syntactic language model – and a hybrid of these.

2.1. N-grams

N-gram language models are based on the idea that each word in a sentence can be assigned a probability of occurrence, based on the preceding words in a sentence, i.e. the following quantity can be estimated:

$$P(w_i|w_1^{i-1}) \tag{2}$$

This allows us to estimate $P(w_1^n)$ as

$$P(w_1^n) = \prod_{i=1}^{n} P(w_i|w_1^{i-1}) \tag{3}$$

N-gram language models are based on a simple counting of the frequencies of strings of words in a training corpus. Since the amount of training data required to estimate $P(w_i|w_1^{i-1})$ increases dramatically as i increases, N-gram models approximate this quantity as

$$P(w_i|w_1^{i-1}) \simeq \hat{P}(w_i|w_{i-N+1}^{i-1}) \tag{4}$$

2.2. SCFGs

A stochastic context free grammar is context free grammar in which all productions are augmented with a probability. Thus the model is defined by (V,T,R,S) where

V = The set of non-terminal symbols
T = The set of terminal symbols
R = The set of (probabalistic) productions
S = The start non-terminal

The model is parameterized by the set of probabilities $P(r)\,\forall r \in R$.

This model allows us to calculate the probability that a given string of terminals, x, is derived from any non-terminal, X (see (Stolcke 1993)):

$$P(X \stackrel{*}{\Rightarrow} x) \tag{5}$$

We can therefore calculate the probability of any word string $P(w_1^n)$ as

$$P(w_1^n) = P(S \stackrel{*}{\Rightarrow} w_1^n) \tag{6}$$

Syntactic language models have the advantage that they are able to model long distance dependencies between words, without the enormous number of parameters that would be required by an N-gram model. Because syntactic models allow for recursive productions such as [**NP** → **NP PP**], they are able to model arbitrarily long sentences without any increase in the number of parameters.

2.3. Hybrid models

Stolcke developed (Stolcke 1993) a probabilistic extension of Earley's parser (Earley 1970) which allows for the calculation of *prefix probabilities*. The prefix probability $P(S \stackrel{*}{\Rightarrow}_L x)$ is the sum of the probabilities of all sentence strings having x as a prefix, and is defined as

$$P(S \stackrel{*}{\Rightarrow}_L x) = \sum_{y \in V^*} P(S \stackrel{*}{\Rightarrow} xy) \tag{7}$$

where

V^* = The set of all possible strings of non-terminals

We can use Equation 7 in a technique which generates N-gram probabilities directly from SCFGs (Stolcke and Segal 1994). This allows us to generate an N-gram language model which contains some knowledge about syntax. There are three reasons why this is desirable:

- It allows us to easily combine our syntactic model with other N-gram models.
- It allows us to integrate our syntactic language model in an existing ASR system for rapid testing.
- Applying an N-gram model is fast – either as a table lookup or directly incorporated into Hidden Markov Model (HMM) transition probabilities.

Of course there are also drawbacks:

- We lose some knowledge about syntactic structure.
- We lose the compactness of our representation.

2.4. Syntactic knowledge in N-grams

Although it may not be immediately obvious, it is possible for an N-gram model which is derived from an SCFG to retain some syntactic knowledge. Consider a trivial training set consisting of two sentences, shown parsed in Figure 2(a) and a test sentence shown parsed in Figure 2(b).

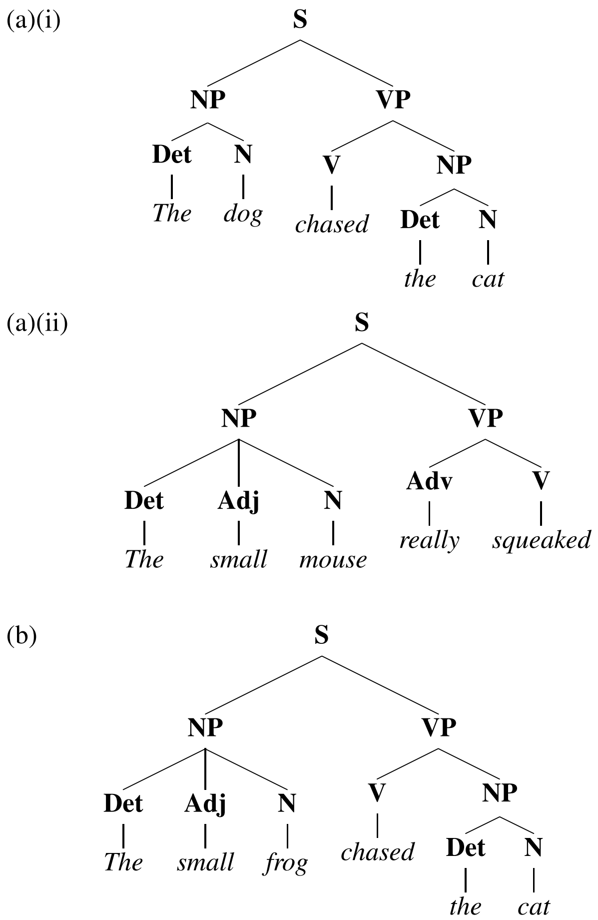

Figure 2: *(a) A trivial training set. (b) A test sentence.*

A trigram model trained on these training examples would not be able to estimate the probability $P(chased_i|small_{i-2}, frog_{i-1})$,[1] since the sequence $\{small, frog, chased\}$ did not appear in the training set. A class-based N-gram approach would at least allow us to estimate the bigram $P(chased_i|frog_{i-1})$ as $P(chased_i|\mathbf{V}_i)P(\mathbf{V}_i|\mathbf{N}_{i-1})$ [2]. This solution is still deficient, however: since there are no training examples containing the class sequence $\{$**Adj**, **N**, **V**$\}$, we are unable to estimate the trigram $P(chased_i|small_{i-2}, frog_{i-1})$. However, because a syntactic model contains higher level structure, it is able to model this sequence. The training examples use the rules [**NP** $\rightarrow$ **Det Adj N**] and [**VP** $\rightarrow$ **V NP**] which can be glued together using the rule [**S** $\rightarrow$ **NP VP**]. Since we have probability estimates for each of these rules obtained from the training data, we can meaningfully estimate the probability $P(chased_i|small_{i-2}, frog_{i-1})$. This is an example of an N-gram that we can estimate using syntactic structure that would not be available otherwise.

[1]Subscripts refer to the position of the word in the sentence

[2]This equation requires that the class of word $i-1$ is known, which implies that each word can only belong to one class.

2.5. Our Syntactic Model

For this work we implemented a full parser based on Stolcke's probabilistic extension to Earley's parser (Stolcke 1993). Since this parser can calculate prefix string probabilities, it is suitable for use as a language model in a speech recognition system. The probability of a hypothesized word, w_n given a word history w_1^{n-1}can be modeled as

$$P(w_n|w_1^{n-1}) = \frac{P(S \overset{*}{\Rightarrow}_L w_1^n)}{P(S \overset{*}{\Rightarrow}_L w_1^{n-1})} \quad (8)$$

We are able to train this model on unparsed text corpora using the EM algorithm. Since this is an initial investigation into the usefulness of modeling corrections syntactically, we transformed our model into a bigram model as described in Section 2.3.. This allows easy testing of our model and integration with the simple bigram model.

3. Correction Modeling

One of the simplest and most commonly employed syntactic corrections is of the form $[X \rightarrow \cancel{X}\ X]$. This represents the process whereby a speaker replaces one syntactic constituent with another equivalent one, e.g. (a) "The man entered the room $\perp$ left the room."[3] in which the verb phrase (**VP**) "entered the room" is replaced by the **VP** "left the room", and (b) "The woman with the coat $\perp$ in the coat smiled." in which the preposition phrase (**PP**) "with the coat" is replaced by the **PP** "in the coat" (See Figure 3). These syntactic corrections pose a problem for a grammar which has been trained on text, as these corrections are not used in formal writing. We must therefore introduce productions to handle these corrections, which we call "correction productions". These productions extend our written grammar to a spoken grammar.

We develop our model as a multi-stage process:

1. An SCFG is initialized on a pre-parsed corpus
2. This SCFG is refined by reestimating its parameters on a large number of (unparsed) training sentences.
3. Correction productions are optionally introduced.
4. The model is reestimated in order to estimate probabilities for correction productions.
5. The SCFG is transformed into a bigram model.

4. Experimental setup

We train our models using sentences from three different sources, listed in Table 2.

[3]The point of disfluency is indicated by '$\perp$'

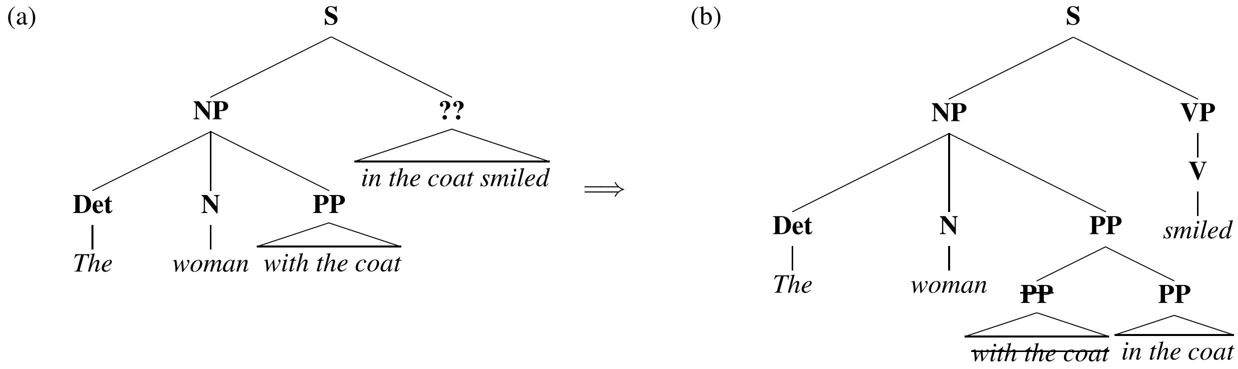

Figure 3: *(a) An "unparsable" sentence. (b) The syntactically corrected form of (a).*

Table 2: *Training sets.*

Training set	Source
$Train_a$	Penn Treebank [103 228 sentences]
$Train_b$	TDT-2 [7000 sentences]
$Train_c$	Switchboard [1000 sentences]

We define three standard bigram models with which to compare our syntactic models. These are listed in Table 3. We also define three class-based bigram models, which are listed in Table 4. We set the number of classes to 13, to match the number of "part of speech" symbols in our syntactic models.

Table 3: *Bigram models.*

Bigram model	Trained on
B_1	$Train_a$ and $Train_b$
B_2	$Train_c$
B_3	$Train_a$, $Train_b$ and $Train_c$

Table 4: *Class-based bigram models.*

Class-based bigram model	Trained on
C_1	$Train_a$ and $Train_b$
C_2	$Train_c$
C_3	$Train_a$, $Train_b$ and $Train_c$

These bigram models use Good-Turing discounting (Good 1953; Church, Gale, and Kruskal 1991).

Our syntactic models are first initialized on $Train_a$, by counting the number of occurrences of each production. Only a subset of $Train_a$, consisting of 24 337 sentences, contains fully parsed sentences; the rest are only annotated with part-of-speech tags and can only be used to estimate "part-of-speech production" probabilities e.g. $P(\mathbf{N} \rightarrow dog)$. Our syntactic models are then retrained on $Train_b$. At this stage we create a number of model variations by optionally augmenting our grammar with "correction productions." We differentiate between *audible* correction productions, in which there is an explicit correction marker, such as "uh", and *inaudible* correction productions in which there is no explicit marker. These model variations are summarized in Table 5. The final step is to retrain on $Train_c$ which contain disfluencies.

Table 5: *Syntactic models.*

Model name	Extra productions introduced
G_0	–
$G_{inaudible}$	$X \rightarrow \cancel{X}\ X \quad \forall X \in V$
$G_{audible}$	$X \rightarrow \cancel{X}\ D\ X \quad \forall X \in V$
	$D \rightarrow$ "*um*"
	$D \rightarrow$ "*uh*"
	$D \rightarrow$ "*well*"
	$D \rightarrow$ "*yeah*"
G_{both}	All productions added to $G_{audible}$ and $G_{inaudible}$.

Each language model uses a vocabulary consisting of all the words occuring in $Train_b$ and $Train_c$, which gives us a vocabulary size of 16 549. We test our models on a set of 1000 sentences from the Switchboard corpus of conversational telephone speech, and a set of 1000 sentences from the TDT-2 corpus of newswire stories, both of which are isolated from the training and development sets. There is a separate development set for the Switchboard and TDT-2 tests, each of 1000 sentences. The performance of each model is measured by its perplexity on the test sets.

4.1. Perplexity

The measure that we refer to as "perplexity" is actually the cross-perplexity and is a measure of how well a language model can "explain" a set of test sentences. It is defined as

$$\begin{aligned}
\text{Perplexity} &= 2^{H(p,m)} \\
\text{where} \quad H(p,m) &= \lim_{n \rightarrow \inf} \frac{1}{n} \sum_{w_1^n \in W} p(w_1^n) \log m(w_1^n) \\
W &= \text{the set of all possible strings in} \\
& \text{a language.} \\
p &= \text{the actual probability distribution}
\end{aligned}$$

		for the language.
m	$=$	the language model being tested, i.e. an approximation to p.

Due the the Shannon-McMillan-Breiman theorem, under the assumption of a stationary and ergodic language, we can simplify this (Jurafsky and Martin 2000) to

$$H(p, m) = \lim_{n \to \inf} -\frac{1}{n} \log m(w_1^n) \quad (9)$$

5. Results

The test results are listed in Tables 6 and 7, where the '$\oplus$' symbol means linear interpolation. Linear interpolation of two bigram models is performed by weighting the first model by λ and the second by $(1 - \lambda)$, where λ is chosen to minimise the perplexity on the developement set.

Table 6: *Test Results on the Switchboard Test Set.*

Model	Perplexity
B_1	693.9
B_2	**259.0**
B_3	269.9
C_1	485.1
C_2	**173.1**
C_3	562.0
G_0	476.6
$G_{audible}$	464.7
$G_{inaudible}$	**452.4**
G_{both}	457.4
$B_2 \oplus G_0$	181.2
$B_2 \oplus G_{audible}$	180.6
$B_2 \oplus G_{inaudible}$	**180.2**
$B_2 \oplus G_{both}$	180.5

Table 7: *Test Results on the TDT-2 Test Set.*

Model	Perplexity
B_1	298.8
B_2	4677
B_3	**297.6**
C_1	**225.1**
C_2	260.0
C_3	232.3
G_0	2137
$G_{audible}$	2066
$G_{inaudible}$	**2039**
G_{both}	2059
$B_3 \oplus G_0$	295.6
$B_3 \oplus G_{audible}$	295.3
$B_3 \oplus G_{inaudible}$	**295.2**
$B_3 \oplus G_{both}$	295.4

6. Discussion

6.1. Switchboard test

We have found that the bigram models trained on the switchboard training data (B_2, C_2) performed better than any other single model on the switchboard test data. The syntactically derived models were competitive however, as they performed better than the bigrams trained on the out of domain data (B_1).

When combined with the best performing standard bigram model via linear interpolation, the syntactically derived models add significantly to the performance. The choice of syntactic model did not significantly affect the outcome, however. This may be due to the fact that the supply of extra bigrams, as described in Section 2.4., is the overriding contribution of the syntactic model in this case, rather than the modeling of corrections. This is supported by the fact that the class based N-grams yielded similar performance. As the amount of data available for training the standard bigram model increases, the importance of this contribution could be expected to decrease, as more of the bigrams previously only available due to the syntactic model are seen in the bigram training data. This contribution would still be important, however, for those bigrams that occur only rarely in the training data. It should be noted that in this experiment we are not exploiting the syntactic models for their ability to model long distance dependencies. This is the aspect of syntactic models in which one would expect to find more complementary information, regardless of the amount of training data available to the bigram model. Longer distance dependencies would be better exploited when higher order syntactically derived N-gram models are used, and the extra N-gram contributions would also be more significant in this case.

By considering only the syntactically derived models, we can observe the effect of modeling disfluencies. As expected, the grammatical model which does not include any correction productions (G_0) performs the worst. This was expected as the grammar is not rich enough to properly model the disfluencies in the switchboard corpus. The results of the grammars with the "audible" ($G_{audible}$) and "inaudible" ($G_{inaudible}$) correction productions are also as expected. The majority of corrections in the switchboard corpus do not have audible correction markers (Shriberg 1996)[4], and this is reflected in the performance of the $G_{inaudible}$ model, which has the best performance of the syntactically derived models. $G_{audible}$ is presumably able to model the rarer corrections that contain an audible correction marker, and so is able to outperform G_0. G_{both} is not quite able to match the performance of $G_{inaudible}$, however its performance may improve with more training data and training iterations.

It could be argued that $G_{inaudible}$ outperforms the other models because it introduces extra flexibility (with respect to G_0) but does not suffer from an excess of parameters that need to be trained, as $G_{audible}$ arguably does. The fact that G_{both} outperforms $G_{audible}$, however, suggests that an excess of parameters is not the only reason for the

[4] Shriberg's filled pause category does not count as a correction according to our definition.

poor performance of $G_{audible}$ (with respect to $G_{inaudible}$). G_{both} has in fact more parameters than $G_{audible}$ but seems to benefit from the "inaudible" productions. Since the difference in performance between the four syntactically derived models is not terribly large, it is difficult to draw definite conclusions, however, the results do suggest that choosing the correct disfluency productions can help on a corpus containing disfluencies.

6.2. TDT-2 test

In this test, the bigram model which was trained on the out of domain data (B_2) performed surprisingly poorly. Interestingly, it performed much worse than the syntactically derived models, which were reestimated on out of domain data. This is likely due to the fact that the training set on which B_2 was trained, $Train_c$, was quite small. Thus, the syntactically derived models gain comparatively more advantage due to their extra bigram generating ability.

Modeling disfluencies should only be useful in this test to the extent that they inadvertently help model fluent speech. Indeed the difference between the relative improvements of the correction grammars ($G_{audible}$, $G_{inaudible}$ and G_{both}) with repsect to G_0 is somewhat less marked in this test than for the switchboard test. However, since there is some improvement, more work must be done to discover exactly what effect the correction productions are having on the syntactic modeling of fluent speech.

When the syntactically derived models were interpolated with the best performing standard bigram, there was not a significant improvement in performance. In this case, the standard bigram model was simply much better than the syntactically derived model. It should be noted, however, that the interpolation did not harm the result.

7. Conclusion

The ability of syntactically derived models to estimate probabilities for word sequences that did not occur in the training data was found to be useful. This was more apparent when the training set was smaller. This effect was not an improvement over class-based N-grams, however it is expected that as the order of N-gram used is increased, the competitiveness of the syntactically derived model would improve, as it would be possible to better exploit the higher-level syntactic structure. This is an avenue of further research.

There appears to be some benefit to modeling corrections as a syntactic process. The syntactically derived models that explicitly modeled the corrections commonly found in the Switchboard database outperformed those that did not. This provides a motivation for further investigating the use of syntactic correction models on interactive speech. However, the conclusions that can be drawn from this are limited at this stage, since a moderate version of the same effect occurred on the TDT-2 database. Further investigation is also required to determine the true contribution of such models when they are used in their original form, rather than being transformed into bigrams.

8. Acknowledgements

The author acknowledges financial support provided by the Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM)2. The NCCR is managed by the Swiss National Science Foundation on behalf of the Federal Authorities.

References

Carbonell, J. G. and P. J. Hayes (1983). Recovery strategies for parsing extragrammatical language. *American Journal of Computational Linguistics 9*(3-4), 123–146.

Church, K. W., W. A. Gale, and J. Kruskal (1991). Appendix A: The Good-Turing theorem. *Computer Speech and Language*, 19–54.

Earley, J. (1970). An efficient context-free parsing algorithm. *Communications of the ACM 6*(8), 451–455.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika 40*, 237–264.

Hindle, D. (1983). Deterministic parsing of syntactic non-fluencies. In *Proc. of 21st Annual Meeting of the Association of Computational Linguistics*, Massachusetss.

Jurafsky, D. and J. H. Martin (2000). *Speech and Language Processing*. Prentice-Hall.

Levelt, W. (1989). *Speaking*. The MIT Press.

McKelvie, D. (1998). SDP – Spoken Dialogue Parser. HCRC Technical Report HCRC/TR-96, Human Communication Research Centre, Edinburgh.

Rosenfeld, R. (1994, Apr). *Adaptive Statistical Language Modelling: A Maximum Entropy Approach*. Ph. D. thesis, School of Computer Science, Carnegie Mellon University.

Schegloff, E. A. (1987). *The micro-macro link*, Chapter 9, pp. 207–234. University of California Press.

Shriberg, E. (1996). Disfluencies in switchboard. In *Proc. International Conference on Spoken Language Processing*, Volume Addendum, Philadelphia, PA, pp. 11–14.

Stolcke, A. (1993). An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. Technical Report TR-93-065, International Computer Science Institute, Berkeley, CA, USA.

Stolcke, A. and J. Segal (1994). Precise n-gram probabilities from stochastic context-free grammars. In *Meeting of the Association for Computational Linguistics*, pp. 74–79.

Young, S. (1996, Sep). A review of large-vocabulary continuous-speech recognition. *IEEE Signal Processing Magazine 13*(5), 45–57.