The Role of Disfluencies in Topic Classification of Human-Human Conversations

Constantinos Boulis, Jeremy G. Kahn and Mari Ostendorf
Signal, Speech and Language Interpretation Laboratory,
Dept. of Electrical Engineering, Univ. of Washington, Seattle, WA 98195
{boulis,jgk,mo}@ssli.ee.washington.edu

Abstract

We investigate the impact of disfluencies on the task of classifying natural human-human conversations into topics. Disfluencies are distinctive to spoken language, and their effect on a number of spoken language understanding tasks, including spoken language classification, remains largely unknown. We use a subset of Switchboard-I annotated for disfluencies and topics, and investigate the effect of different disfluency categories with both true and automatically generated transcripts. We show that under the popular bag-of-words representation, even perfect disfluency filtering has a minimal impact on topic classification performance on hand-transcribed data. However, difference are larger with more complex representations (e.g. bigrams) and for some classifiers operating on recognizer transcripts.

Introduction

Classifying human-human conversations to topics can be an important part in a number of applications ranging from analyzing business meetings to customer call-centers. Moreover, conversation classification shares a number of issues with spoken language understanding tasks, such as call-routing. As in call-routing, natural, spontaneous speech is mapped to a single topic. But unlike call-routing, the conversations are longer than a few sentences and there is interaction between two parties. Dealing with spontaneous speech brings forward a wide array of issues, such as converting speech to text, utilizing prosodic aspects of the speech signal, and investigating the effects that pronouns and disfluencies have on the classification performance. Although a number of approaches have been suggested for converting speech to text for call routing, for example using a word recognizer and compensating for errors (Tür *et al.* 2002; Siegler & Witbrock 1999), the rest of the issues have not been extensively studied.

In this work, we investigate the effect disfluencies have on conversation classification performance. Disfluencies occur amply in spoken language (Shriberg 1994), and although at the surface they appear to interrupt the flow of information, human listeners typically have little trouble understanding disfluent speech. For automatic language processing though, disfluencies falsely increment the counts of words, and since the most prevalent representation for topic classification is the bag-of-words, they can potentially have an adverse effect on conversation classification. In the past, attempts have been made to detect disfluencies in conversations (Liu, Shriberg, & Stolcke 2003). Removal of disfluencies has been shown to increase the readability of conversation transcripts (Jones *et al.* 2003) and detecting and removing repetitions, a certain type of disfluency, has been used to produce more natural summaries of spoken dialogues (Zechner & Waibel 2000). In addition, handling of disfluencies is important at the grammar component of an SLU system (Wang 1999).

In this work, we decompose disfluencies to five categories, similar to (Shriberg 1994), and study the effect of different groups of them. The five categories, with an example for each, are shown below:

- **Fillers**. *Uh, well like, one week she'll work three days and I'll work two* . In this example, we see two kind of fillers: filled pauses (*uh*) and discourse markers (*well*,*like*).
- **Restarts**. *I have to plan way in advance, because, + or, what I've done is found like doctors' and dentists' office with extended hours*. The + sign marks the starting point of the new sentence.
- **Repairs**. *And, uh, I called you know from [[that, + the,] + the] T I Data Base Calling Instructions*. This is an example of nested disfluencies.
- **Repeats**. *Plus, I bet it [cuts, + cuts] down on your absenteeism*.
- **Word Fragments**. *Yeah, but I can [rem-, + remember] back growing up*.

It is possible that some categories may have no effect on topic classification performance, while others negatively impact performance. For example, fillers, such as filled pauses and discourse markers, are very frequent in a conversation, so their relevance (or lack thereof) should be robustly estimated using the text with disfluencies. Restarts and repairs represent a more interesting category since the intention of the speaker changes (or is repaired) and this may adversely affect performance. Repeats distort counts, but the majority are on very frequent words *I I am sure...*, so the argument for the filler category may apply. Repeats are easiest to detect, so it would be good news if they dominate any

Copyright © 2005, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

performance differences due to being the most frequent category. Finally, word fragments are represented as separate tokens - when the true transcripts are used - therefore increase the vocabulary size. Since all statistical topic classification algorithms employ smoothing techniques, increasing the vocabulary size may have the deleterious effect of moving more probability mass from relevant to irrelevant words. Moreover, when using an ASR system, word fragments will always be erroneously mapped to a full word or deleted, possibly impacting the neighboring words as well.

Research Questions

There are four main questions that we answer in this work:

- **Does the removal of disfluencies lead to a better document representation for topic classification?** We include experiments using a variety of classifiers to verify that there is a consistent improvement of performance. In addition, we investigate the effect of disfluencies on the bag-of-words and bag-of-word-pairs representations, since the impact of disfluencies may depend on the choice of representation. Finally, we look at different classes of disfluencies, since some are easier to automatically detect than others.
- **How do disfluencies interact with feature selection?** Many of the words in a disfluent segment are high frequency and not closely-associated with any topic, such as *I mean*, *um* etc. It is possible that feature-selection methods remove most of the words from disfluent regions of the text. Alternatively, it may be that removing disfluencies before feature selection leads to better results.
- **Can feature selection be improved by first removing disfluencies?** In standard feature selection methods, all occurrences of a word are removed from the data. Therefore, if a word is irrelevant in one context and relevant in another, it will still be removed if the aggregate statistics deem it irrelevant. For example, a very common word within a disfluency is *mean*. It can be the case that if the word *mean* is found outside a disfluency it can be relevant — the speaker may be talking about *arithmetic mean* or how *mean* a person is.
- **Do disfluencies impact true transcripts differently than ASR-generated transcripts?** It is possible that disfluencies will have a different impact on topic classification performance when using an ASR system. A word fragment will never be recognized as such using an ASR system. In addition, a disfluency — even without word fragments — may be more challenging to correctly recognize, since speech within a disfluency tends to be less clearly articulated. If more ASR errors happen inside a disfluency than outside, then removal might improve overall topic classification.

The outcome of these experiments can reveal new directions for further research. For example, if removing disfluencies is important for topic classification, can an automatic disfluency detection system be used and with what modifications, e.g. confidence outputs?

Corpus & Task

For all our experiments we have used the Switchboard-I corpus (Godfrey, Holliman, & McDaniel 1992). Switchboard-I was developed in the early 90's and has been mainly used for ASR research. The corpus consists of 5-minute telephone conversations between people who have not met each other before. The topic of a conversation is suggested to the participants prior to the conversation and a *topicality* label, i.e. a label indicating how closely the participants stayed on the suggested topic, is available. A subset of the Switchboard-I corpus annotated for disfluencies is converted from the older TB3 data (Meteer & et al. 1995) to the more recent LDC V5.0 (Strassel 2003), maintaining the correction information. There are in total 1126 conversations or 2252 conversation sides annotated for disfluencies, consisting of about 1.45M term occurrences. The annotation defines three parts for each edit disfluency, the deletable portion, the interruption point and the correction. The deletable portion is the disfluent part of the utterance and the one that gets deleted, the interruption point marks the boundary between the deletable portion and the correction which can involve an editing term or no terms at all, and the correction is the fluent part of the utterance and the one that is retained. An example annotation is shown below, where the deletable portion (DEL) is within square brackets, the interruption point (IP) is marked with the plus sign and the correction (CORR) is within curly braces. The editing term of the correction is shown as EET (explicit editing term).

$$qualifications \underbrace{[that]}_{DEL} \underbrace{+}_{IP} \underbrace{you\ know}_{EET} \underbrace{\{that\}}_{CORR} you\ have$$

It should be noted that edit disfluencies can be overlapping or nested. The annotation methodology does not distinguish between repairs, restarts or repeats. To distinguish the three categories we applied the following simple rules, shown in Algorithm 1.

```
1: if CORR == ∅ then
2:     DISFLUENCY=RESTART;
3: else if DEL == CORR then
4:     DISFLUENCY=REPEAT;
5: else if DEL != CORR then
6:     DISFLUENCY=REPAIR;
7: end if
```

Algorithm 1: Set of rules used to characterize a disfluency as a restart, repeat or repair.

A single topic (from a list of 67) was suggested to the participants before the start of each conversation. The task is to classify a conversation side to one of 67 possible topics. The distribution of conversations to topics is quite imbalanced. The highest number of conversation sides for a topic was 70, the lowest 4 and the median was 34. In all experiments, words with 2 or more occurrences in the entire corpus (train and test) have been retained. This resulted in vocabularies of 13866 and 13192 terms when using text before and after removal of disfluencies respectively. Instead of choosing a specific train and test set, we performed a 10-fold cross

validation test and report the average and standard deviation of results. This allows us to observe the sensitivity of the results to different train/test data.

Methods

We have used two toolkits that are publicly available for research purposes and have implementations of six different text classifiers. The Bow toolkit (McCallum 1996) was used for training five out of six classifiers and the *SVMLight* toolkit (http://svmlight.joachims.org/) was used for training the Support Vector Machines classifier. Both toolkits are popular within the text classification community and have been extensively used in the past. The six classifiers we have used are:

- **Maximum Entropy** (MaxEnt) (Nigam, Lafferty, & McCallum 1999)
- **k Nearest Neighbors** (kNN) (Manning & Schütze 1999)
- **Support Vector Machines** (SVM) (Joachims 1999)
- **Naive Bayes with shrinkage** (NBShrinkage) (McCallum *et al.* 1998)
- **tfidf/Rocchio** (Rocchio) (Joachims 1997)
- **Probabilistic Indexing** (PrIndex) (Fuhr 1989).

We will very briefly describe the last three, lesser-known text classifiers. Naive Bayes with shrinkage is the Naive Bayes classifier with an alternative way of smoothing. Instead of using Laplace smoothing, i.e. if $N(w,c)$ is the count of word w in topic c, we set $\tilde{N}(w,c) = N(w,c)+1$, the topic-specific word distributions are smoothed with the word distribution in the whole training corpus, i.e. $\tilde{p}(w|c) = \lambda p(w|c)+(1-\lambda)p(w)$. Therefore the probability of observing document $\vec{d}$ is given by:

$$p(\vec{d}) = \sum_{c=1}^{C} p(c) \prod_{k=1}^{N^d} (\lambda p(w_k = w|c)+(1-\lambda)p(w_k = w))^{N_w^d} \quad (1)$$

where N_w^d is the number of occurrences of word w in document $\vec{d}$ and N^d is the number of unique words of document $\vec{d}$. The tfidf/Rocchio classifier represents each document m with a weight vector whose k-th element is given by:

$$d_k^m = \frac{f_k^m \log(N_D/n_k)}{\sum_{j=1}^{V} f_j^m \log(N_D/n_j)} \quad (2)$$

where N_D is the number of documents, n_k the number of documents in which the indexing term appears, and f_k^m is the frequency of term k in document m. The representation of class c is then constructed as:

$$\vec{u}_c = \frac{\alpha}{|R_c|} \sum_{m \in R_c} \vec{d}^m - \frac{\beta}{|\bar{R}_c|} \sum_{m \in \bar{R}_c} \vec{d}^m \quad (3)$$

where R_c is the set of training documents of class c and $\bar{R}_c$ is the set of training documents of every class but c. The parameters α and β are tuned either by hand or using cross validation. During testing, a new document $\vec{d}$ is assigned to the class with the maximum cosine similarity:

$$\hat{c} = arg\max_c cos(\vec{d}, \vec{u}_c) \quad (4)$$

The probabilistic indexing classifier is a statistical classifier where a new document d is classified to class $\hat{c}$ according to:

$$\hat{c} = arg\max_c \sum_w p(c|w)p(w|\vec{d}) \quad (5)$$

and $p(c|w)$ is evaluated through Bayes Rule.

For all our experiments the default settings for each classifier have been used. For example, the smoothing coefficient in NBShrinkage was set to $\lambda = 0.6$ and for kNN $k = 30$. For the SVM training, since SVMs are inherently binary classifiers and *SVMLight* does not have implemented multi-class approaches to classification, we used the one-vs-one approach. In the one-vs-one approach, given a C-category classification problem, $C*(C-1)/2$ binary classifiers are constructed for every pair of classes. For each pair $\{i,j\}$ a function $H_{ij}(\vec{d})$ is estimated. During testing, if $H_{ij}(\vec{d}) > 0$ then $votes(i) = votes(i)+1$ else $votes(j) = votes(j)+1$. Document d is assigned to the class with the maximum number of votes $\hat{i} = argmax_i votes(i)$. No attempt to optimize the weight assigned to the training error has been taken. This value is set to default as the variance of the training data.

For the feature selection experiments, we have used the Information Gain (IG) method. IG is a popular filter feature selection method used for text classification that attains very good performance (Forman 2003). IG is given by:

$$IG(w) = H(\mathbf{C}) - p(w)H(\mathbf{C}|w) - p(\bar{w})H(\mathbf{C}|\bar{w}) \quad (6)$$

where $H(\mathbf{C}) = -\sum_{c=1}^{C} p(c)\log p(c)$ denotes the entropy of the discrete topic category random variable **C**. Each conversation side is represented with the Bernoulli model, i.e. a vector of 1 or 0 depending if the word appears or not in the conversation side. Under this representation, $w, \bar{w}$ denote the events of word being present or absent respectively. Words are ranked according to IG and the top N are retained.

Experiments

We have distinguished seven cases in our data. Using the original text with all the disfluencies which we annotate on the tables with **Keep All**, removing all five categories of disfluencies (**Remove All**) and then individually removing each one of the five categories.

Effect of disfluencies on the BOW representation

We begin the experiments using the standard bag-of-words representation. In Table 1, we see the topic classification accuracy across different classifiers, and also by individually removing each disfluency category. The standard deviation of all classification experiments is also reported.

From Table 1, we can see that overall there is a small but consistent difference by removing all disfluencies. Looking at the top 3 results, we find that the differences between **Keep All** and **Remove All** are significant for PrIndex and NBShrinkage ($p < 10^{-3}$) and marginally significant for

	Keep All	Remove All	Remove Fillers	Remove Restarts	Remove Repairs	Remove Repeats	Remove Word Fragments
MaxEnt	78.0±0.9	79.0±0.9	78.4±0.8	78.0±0.8	78.0±0.9	78.3±0.9	78.2±0.8
kNN	83.9±0.6	84.7±0.8	84.0±0.7	84.5±0.5	84.6±1.2	84.4±0.7	84.2±0.5
SVM	83.0±0.4	83.4±0.6	83.4±0.6	83.0±0.5	83.1±0.9	82.6±0.7	83.8±0.6
PrIndex	82.8±2.3	85.6±1.8	84.2±1.1	84.4±1.6	83.5±2.1	84.7±2.0	83.9±1.7
NBShrinkage	91.4±0.9	91.9±0.8	91.8±0.6	91.4±0.4	91.6±0.6	91.6±0.5	91.6±0.5
Rocchio	92.4±2.2	93.1±0.6	93.2±0.6	92.3±1.9	91.4±2.6	92.3±2.2	91.5±2.3

Table 1: *Topic classification accuracy of various classifiers using unigrams as features.*

PrIndex ($p = 0.11$), but not significant for Rocchio, using a Student's t-test on 50 cross-validation subsets in each case. Removing individual disfluency categories provides classification accuracies within the range of two extremes (**Remove All** and **Keep All**). Since the difference between the two extremes is small, it is hard to say what is the relative influence of each one of the categories, but it is certainly the case that it is not a single category that accounts for all of the difference. In Table 2 the relative reduction of word occurrences compared to retaining all disfluencies is shown. The biggest category is fillers which can explain why the impact of removing only fillers appears to be slightly bigger than other categories. Note also that since disfluencies can be nested the sum of words removed from each one of the categories can be higher than the words removed from all categories.

Effect of disfluencies on the BOWP representation

The next question we attempt to answer is whether more complex representations can benefit more from removing disfluencies. Previous work has shown that bigrams can perform better than unigrams for Switchboard-like conversations, when enough training data are available (Boulis & Ostendorf 2005). A reason for this is that bigrams can capture expressions that are inadequately modeled with unigrams. For example, for the topic *"reality shows"* a relevant bigram is *"big brother"*. But neither *"big"* or *"brother"* as individual words can capture this. If disfluencies can disrupt the sequence of such relevant bigrams, for example *big uh, um, brother* then they can effectively weaken the representational capacity of bigrams. In Table 3 we report the results of using bigrams (bag-of-word-pairs or BOWP) as the representation method. Overall, we notice that the difference between the **Keep All** and **Remove All** cases is increased, compared to unigrams (except for PrIndex). This difference (between **Keep All** and **Remove All**) is significant for the Rocchio classifier ($p < 6 \times 10^{-3}$) and for NB-Shrinkage, but not significant for the best-performing classifier (probabilistic indexing). For all classifiers, except probabilistic indexing, it appears that using bigrams degrades the performance considerably compared to unigrams. Surprisingly, probabilistic indexing gets a significant boost, offering the best result over all classifiers and over all representations. Another interesting observation is that the kNN classifier benefits significantly by removing disfluencies when using bigrams as features.

	Keep All	Remove All
MaxEnt	-1.1	-1.1
kNN	-0.2	-0.7
SVM	-1.9	-1.8
PrIndex	+3.3	+1.0
NBShrinkage	-0.2	-0.9
Rocchio	+0.4	+0.1

Table 4: *The effect of feature selection on text with and without disfluencies, using the top 5K unigrams selected with information gain.*

Feature selection and disfluencies

The next two questions we explore are a) whether the negative contribution of disfluencies can be mitigated with feature selection (e.g. words frequently associated with disfluencies are removed in the feature selection process) and, alternatively, b) whether feature selection is more effective when disfluencies are removed. We performed feature selection, training and testing of classifiers on disfluent text and on text with disfluencies removed. Table 4 shows the difference in average classification performance between these two experiments and the first two columns of Table 1. In Table 4, column 1, we can see that keeping only the top 5K IG words does not improve the results compared to using all the word features, except for the case of PrIndex which records a significant boost. This IG-based feature selection removed 60% of the disfluency word types and 74% of the disfluency tokens, so it appears that there are topically important words in disfluency regions and that these impact performance. To answer the second question, we can compare the columns of Table 4, which show that IG-based feature selection is not improved by using text with disfluencies removed.

ASR-generated transcripts and disfluencies

We have used the SRI Decipher ASR system (Stolcke & et al. 2004) to decode all the 2252 conversation sides. We have used only the first step of the entire decoding process which consists of using bigram language models with unadapted MFCC acoustic models to perform the decoding. Since these data have been part of the training data of the SRI Decipher system, continuing for subsequent decoding steps would result in an unrealistically low word error rate (WER). The WER using only the first step is 30.2%. Since we have available the time segments of each disfluency and

Keep All	Remove All	Remove Fillers	Remove Restarts	Remove Repairs	Remove Repeats	Remove Word Fragments
-	11.9%	6.6%	0.8%	2.8%	2.6%	0.9%

Table 2: *Relative reduction of word counts from removing different disfluency categories.*

	Keep All	Remove All	Remove Fillers	Remove Restarts	Remove Repairs	Remove Repeats	Remove Word Fragments
MaxEnt	63.4±1.1	65.4±1.3	64.1±0.8	63.5±1.2	63.6±0.7	63.0±0.7	62.1±0.8
kNN	72.8±1.1	79.2±0.7	78.0±1.0	72.8±0.8	74.0±1.0	74.1±0.9	73.1±1.0
SVM	49.6±1.4	50.8±1.5	50.4±1.8	49.8±0.9	50.2±1.1	50.3±0.9	50.0±1.2
PrIndex	94.3±0.4	94.5±0.7	94.4±0.5	94.2±0.6	94.5±0.4	94.0±0.5	94.2±0.5
NBShrinkage	81.5±0.4	83.4±0.7	82.6±0.7	82.0±1.0	81.8±0.8	81.8±0.7	81.4±0.5
Rocchio	85.2±0.6	86.4±1.0	86.0±0.7	85.3±0.8	85.8±0.7	85.4±0.7	85.5±0.9

Table 3: *Topic classification accuracy of various classifiers using bigrams as features.*

	Keep All	Remove All
MaxEnt	76.6±1.1	78.1±0.8
kNN	83.9±0.3	83.1±0.7
SVM	81.7±0.6	82.4±0.3
PrIndex	81.7±1.5	84.3±1.0
NBShrinkage	89.9±0.5	89.9±0.7
Rocchio	85.1±1.4	91.5±0.9

Table 5: *Topic classification on the ASR transcripts using unigrams as features.*

the SRI Decipher system outputs the time segments for each word, we can remove all words that fall mostly within a disfluency. Here, "mostly" refers to more than half of the word's duration being within a disfluency. This process is not perfect: words that should not have been removed will be removed, and words that should have been removed will not be removed, but it is fairly accurate.

The results are shown in Table 5 and show a mixed picture. The best classifier in other experiment, Rocchio, degrades substantially with ASR errors (comparing with Table 1) and clearly benefits from removing disfluencies. The second best classifier, NBShrinkage, degrades somewhat with ASR but is not helped by disfluency removal. For most classifiers, there is little degradation due to ASR (despite a 30% WER) and a small benefit to disfluency removal, but it may be that any benefit is lost (or worse) with automatic disfluency detection. Hence, proper choice of classifier is more important than disfluency removal with ASR transcripts.

Discussion

In this work we have explored the impact of disfluencies on topic classification of natural human-human conversations. Overall, we can say that removing disfluencies has a small impact on the bag-of-words representation but appears to have a bigger impact on the bag-of-word-pairs representation. Another observation from our experiments was that feature selection can not effectively remove disfluencies. In addition, feature selection does not appear to be greatly improved by first removing disfluencies. Lastly, we have explored the effect of disfluencies on ASR-generated transcripts. On ASR transcripts, we have found that the effect of both word errors and disfluency removal is highly dependent on the classification method, with the greatest benefit from disfluency removal coming for the classifier most sensitive to errors. For both true and ASR transcripts the best performance is achieved by removing disfluencies, but the relative gain is not large and may be lost with automatic disfluency detection. Overall, we find that choice of classifier has a much bigger effect than disfluency removal. With current classifiers and the bag-of-word representation, there appears to be little need for disfluency removal, though this could change if future developments make more use of word sequence patterns.

The conclusions of this study need to be interpreted with the caveat that the corpus was explicitly designed to include dialogs on a single topic. In multi-topic and.or multi-party speech, disfluencies may play a more important role, and similarly for more fine-grained topic labeling. In addition, there are other task in language processing where disfluencies might be informative (rather than interpreted as noise), such as speaker and topic segmentation.

Acknowledgments

The authors wish to thank Elizabeth Shriberg for suggesting the analysis by disfluency type and anonymous reviewers for their suggestions. This work was supported by NSF grant IIS-0121396. Any opinions, findings, and conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the funding agency.

References

Boulis, C., and Ostendorf, M. 2005. Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. In *Proc. of the International Workshop in Feature Selection in Data Mining*, in press.

Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research* 1289–1305.

Fuhr, N. 1989. Models for retrieval with probabilistic indexing. *Inf. Process. Manage.* 25(1):55–72.

Godfrey, J.; Holliman, E.; and McDaniel, J. 1992. Switchboard: Telephone speech corpus for research development. In *Proceedings of ICASSP*, 517–520.

Joachims, T. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proc. of ICML*.

Joachims, T. 1999. *Making large-Scale SVM Learning Practical*. MIT-Press.

Jones, D.; Wolf, F.; Gibson, E.; Williams, E.; Fedorenko, E.; Reynolds, D.; and Zissman, M. 2003. Measuring the readability of automatic speech-to-text transcripts. In *Proc. of Eurospeech*.

Liu, Y.; Shriberg, E.; and Stolcke, A. 2003. Automatic disfluency identification in conversational speech using multiple knowledge sources. In *Proc. of EuroSpeech*.

Manning, C., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

McCallum, A.; Rosenfeld, R.; Mitchell, T.; and Ng, A. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of ICML*.

McCallum, A. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow.

Meteer, M., and et al. 1995. Disfluency annotation stylebook for the Switchboard corpus. In *Linguistic Data Consortium*.

Nigam, K.; Lafferty, J.; and McCallum, A. 1999. Using maximum entropy for text classification. In *Proc. of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, 61–67.

Shriberg, E. 1994. *Preliminaries to a theory of speech disfluencies*. Ph.D. Dissertation, University of California, Berkeley.

Siegler, M., and Witbrock, M. 1999. Improving the suitability of imperfect transcriptions for information retrieval from spoken documents. In *Proceedings of ICASSP*, 505–508.

Stolcke, A., and et al. 2004. SRI system description. In *EARS RT04 Workshop*.

Strassel, S. 2003. Simple metadata annotation specification version 5.0.

Tür, G.; Wright, J.; Gorin, A.; Riccardi, G.; and Hakkani-Tür, D. 2002. Improving spoken language understanding using word confusion networks. In *Proceedings of ICSLP*, 1137–1140.

Wang, Y. 1999. A robust parser for spoken language understanding. In *Proc. of Eurospeech*.

Zechner, K., and Waibel, A. 2000. DIASUMM: Flexible summarization of spontaneous dialogues in unrestricted domains. In *Proceedings of COLING*, 968–974.