

A Three-stage Disfluency Classifier for Multi Party Dialogues

Margot Mieskes¹, Michael Strube²

¹European Media Laboratory GmbH, Heidelberg, Germany
<http://www.eml-d.de/english/homes/mieskes>

²EML Research gGmbH, Heidelberg, Germany
<http://www.eml-research.de/~strube>

Abstract

We present work on a three-stage system to detect and classify disfluencies in multi party dialogues. The system consists of a regular expression based module and two machine learning based modules. The results are compared to other work on multi party dialogues and we show that our system outperforms previously reported ones.

1. Introduction

Disfluencies are a very common phenomenon in spoken language¹. Disfluencies have been described linguistically by Shriberg (1994) and Lickley (2001). The latter focuses on causes for hesitations and self-repairs and the effects on disfluency rates. Shriberg (1994) proposes a characterization of the major classes of disfluencies and gives a comprehensive overview on the phenomenon based on two-party dialogues.

Several attempts to automatically detect and classify disfluencies have been made in the past. These approaches can be grouped into three categories: speech first, transcription based and syntactic approaches.

Speech first approaches use acoustic and prosodic correlates of repair events, but the clues (e.g. F_0 increase) did not help the detection of speech repairs (Nakatani & Hirschberg, 1993). Stouten & Martens (2004) and Pakhomov & Savova (1999) report that the benefit of using speech recognition is small or not significant. Snover et al. (2004) and Heeman & Allen (1999) present two approaches based on transcribed speech. The first used only lexical features, whereas the second also use prosodic features. Their disfluency detection was incorporated into the part-of-speech tagging process and based on decision tree learning. Work on syntactic/parsing approaches include work by Johnson & Charniak (2004), Core & Schubert (1999) and Lendvai et al. (2003) among others. Johnson & Charniak (2004) use a tree-adjointing grammar. Core & Schubert (1999) use a parser on task oriented human-human dialogues. Lendvai et al. (2003) uses a k-nearest neighbor learning mechanism.

Liu et al. (2005) compare an HMM model, a maximum entropy model and a conditional random field model on human transcripts and speech recognition output. The data they use are telephone conversations and broadcast news speech. They found that maximum entropy and conditional random fields perform better than the HMM model.

Work on multi party dialogues concentrates on discourse markers (“well”, “like” etc.) and various approaches to their detection (Popescu-Belis et al., 2004). Baron et al. (2002) compare effects of speech recognition on disfluency detection and found that the prosody-based model is

more robust with a real speech recognition input. Zechner (2001) did work on dialogues and also multi party dialogues. He used a three-stage approach, based on Part-of-Speech (POS) tagging and machine learning.

Our work is based on results reported by Shriberg (1994), Lickley (2001) and Zechner (2001), to which we will also compare our results.

2. Disfluency Types and Manual Annotation

The *ICSI Corpus* contains 75 meetings (Janin et al., 2003). We chose 12 meetings randomly to be annotated by two human annotators, to assess the inter-annotator agreement. One meeting was used to train the human annotators and another 26 meetings were manually annotated individually by the annotators. From the total of 38 manually annotated meetings, three were used for testing and the remaining 35 were used for training. Based on previous work we distinguish the following types of disfluencies:

nlfp nonlexicalized filled pauses (e.g. *uh, um, ah*)

lfp lexicalized filled pauses (e.g. *like, well*)

repa repairs (e.g. *Well, they - they have s- they have the close talking microphones for each of us*)

repet verbatim repetitions (e.g. *I know you were - you were doing that*)

abw abandoned words (e.g. *w-, h-, shou-*)

abutt abandoned utterances (e.g. *the newest version after your comments, and-*)

The inter annotator agreement is $\kappa = 95.2$. This result shows, that the annotation was performed very reliably. This agreement was determined on a token basis, as the smallest unit that can be a disfluency are tokens. The annotation was performed iteratively: first one-token disfluencies (e.g. NLFP) were annotated and afterwards two-token disfluencies and so on. Table 1 shows the frequency of tokens belonging to one of the categories classified.

In order to determine the quality of single categories in the manual annotation, a method based on κ was used, which uses κ_j , where j is the j^{th} category. Fleiss (1971)

¹The work reported in this paper was done while the first author was affiliated with EML Research gGmbH.

type	relative
NLFP	23.6
repet	14.5
LFP	23.4
abw	7.0
repa	17.9
abutt	13.5

Table 1: Relativ frequencies of disfluency types

extended the original κ (Cohen, 1960) for k categories and m raters (see also Siegel & Castellan (1988, pp.284-291)). Its calculation is very similar to κ :

$$\kappa_j = \frac{\bar{P}_j - p_j}{1 - p_j} = \frac{\sum_{i=1}^N n_{ij}^2 - N \cdot m \cdot p_j \cdot (1 + (m - 1) \cdot p_j)}{N \cdot m \cdot (m - 1) \cdot p_j \cdot q_j}$$

where N is the number of examples, $q_j = 1 - p_j$ and p_j is

$$p_j = \frac{\sum_{i=1}^N n_{ij}}{N \cdot m}.$$

The results using this formula are similar to those obtained by the method applied by Teufel & Moens (2002), where all categories except of the one of interest are mapped to one and κ is calculated. This indicates that the intuition behind this method is the same, but the formula gives a more straightforward way to get the results.

Applying this formula to the manually annotated data gave the results presented in Table 2. As can be seen, some categories achieve a lower κ_j result than others. Especially *repa* and *abutt* are considerably lower than the other categories. This indicates that these categories are harder to distinguish than the others.

	κ	κ_{nlfp}	κ_{lfp}	κ_{repet}	κ_{repa}	κ_{abw}	κ_{abutt}
total	95.2	99.6	97.7	98.2	78.3	96.0	85.3

Table 2: κ and κ_j results for the manual annotation on segments

3. Zechner’s Approach

Zechner (2001) used several corpora which included a set of group meetings which were recorded in the Interactive Systems Labs at Carnegie Mellon University. The main differences are that the topics in these meetings were predetermined and that the meetings were shorter (in average 304 sentences). The author reports that 13.9% of all sentences are false starts and that 13.2% of all words are disfluent. About 0.87% disfluencies occur per sentence, of which repairs are 29.0%, Nonlexicalized filled pauses are 29.5% and lexicalized filled pauses are 13.9%.

Zechner (2001) used a three component method to detect disfluencies. The first stage is a Part-of-Speech tagger, based on the Brill Tagger (Brill, 1994). Three tags were introduced to cover disfluency phenomena: CO for coordinations, DM for discourse markers (which are treated as lexicalized filled pauses) and ET for editing terms. The tag UH for nonlexicalized filled pauses is a standard tag in the Penn Treebank tagset as used for Switchboard (Godfrey et al.,

1992). The tagger was trained on the manually annotated Switchboard data and tested on a testset also from Switchboard. The results for the Group Meeting data show that coordinations achieve an f-measure of 0.54, discourse markers achieve an f-measure of 0.30, editing terms achieve an f-measure of 0.88 and nonlexicalized filled pauses achieve an f-measure of 0.45.

The second stage deals with false starts. A decision tree was trained on Switchboard data to detect false starts. The features used were trigger words, POS tags and chunks from a chunk parser. Additionally, the length of the sentence in words and number of words not parsed by the chunker. The result achieved on the group meetings data is an f-measure of 0.557.

The final stage is a repetition detection. A script identified repetitions of word/POS sequences of up to four items. Zechner states that longer repetitions only account for less than 1% of all repetitions. Items marked as disfluent in the previous stage were ignored. The f-measure for this method on the group meeting data was 0.41.

As this is to our knowledge the only work that deals with the detection of disfluencies in multi party dialogues we decided to use it as a reference, although the data is only roughly comparable.

4. Automatic Classification

Following Zechner (2001) we set up a multi-stage approach for detecting disfluencies. The Gold Standard Data contains approximately 183,000 tokens. Of these about 43,400 tokens belong to some kind of disfluency (23.7%).

4.1. Regular Expression Based Detection

The regular expression based detection and classification uses different information depending on the disfluency class to be detected. Non-lexicalized filled pauses can be detected based on a list of words like *uh*, *um* etc, but also on the POS tag UH. In order to avoid errors based on annotation errors we used both features. Abandoned words are marked in the transcription with a dash (“-”), without space between the letters and the dash.

Verbatim repetitions are slightly more complicated. Unlike Johnson & Charniak (2004) and Zechner (2001) we did not limit the length of the repetition. Therefore, repetitions can potentially be half as long as the utterance in which they are found. The detection process itself works iteratively: first, all one word repetitions are checked by comparing every word with its neighbour. Second, all two-word repetitions are checked by comparing pairs of words with the neighbouring pairs. This process is repeated up to half the length in words (without interpunctuation) of the utterance in question.

Additionally, as disfluencies are sometimes embedded in disfluencies the detection process first searches for NLFP, next for ABW. These two types are then removed. In the next step one-item repetitions are detected and removed. Then, two-item repetitions are detected and removed, and so on.

As the other three disfluency categories (REPAI, LFP and ABUTT) are not easily detectable with these methods, this part will focus on NLFP, ABW and REPET.

DisflType	prec	rec	f
NLFP	89.56	98.66	93.89
REPET	74.64	93.36	82.95
ABW	89.99	99.19	94.37

Table 3: Regular expression based classification

Table 3 shows results for the regular expression based detection of the three disfluency types NLFP, ABW and REPET. The results show that these three types can be reliably detected by the above described method. REPET is slightly worse than the other two types, but still achieves a very high detection rate.

Figure 1 illustrates the procedure in general.

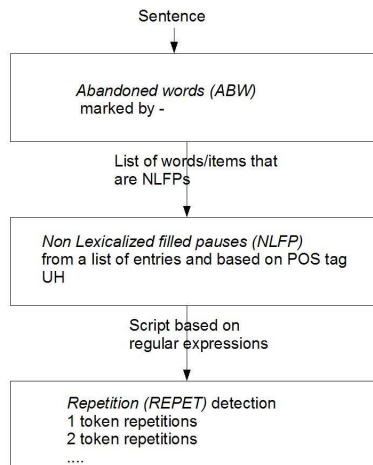


Figure 1: Illustration of the process to detect ABW, NLFP and REPET

Figure 2 presents a simple example of the procedure based on a sentence from the corpus.

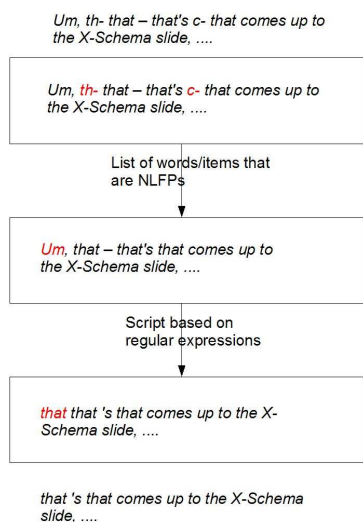


Figure 2: Example for the regular expression based classification for ABW, NLFP and REPET

4.2. Machine Learning Approach

For the machine learning experiments we used a set of features, which were inspired by the literature on disfluencies, mainly Shriberg (1994) and Lickley (2001). Among these features were

- part-of-speech tag,
- length of the utterance considered,
- gender of the speaker,
- native or non-native speaker,
- position of the current utterance in the meeting,
- position of the considered item in the utterance, and
- talkativity features like average length of segments, number of segments uttered, etc.

The whole set of features contained 17 to 21 features. In some experiments presented below we added features based on knowledge about previous disfluencies (see below). The set of features and data from the manual annotation was used to train a decision tree learner (J48 from the Weka Machine Learning Environment²). We did not use features from the speech signal, but only features from the transcription. It has been suggested in the past, that for example longer utterances tend to have relatively more disfluencies (Shriberg, 1994) and that male speakers tend to produce more disfluencies than females. Additionally, Shriberg (1994) observed that certain disfluencies occur more often in sentence initial position than sentence middle or final positions. Therefore, we used the position of the token to be analyzed as a feature.

The first of this set of experiments aimed at classifying which items in the dialogues belong to some kind of disfluency and which do not.

type	accuracy	prec	rec	f
non oversampled				
disfluent	88.5	75.3	55.8	64.1
non-disfluent		90.6	95.9	93.1
oversampled				
disfluent	84.3	61.9	70.2	65.8
non-disfluent		91.5	88.1	89.8

Table 4: Binary classification with no filtering

Table 4 shows the results on the binary classification of disfluent and non-disfluent items. The detection of non-disfluent items is quite successful. Due to the large difference in positive and negative examples we oversampled the training data to achieve an equal distribution of both types. But the results did not improve significantly. The recognition of non-disfluent items dropped considerably.

In order to reduce the amount of possible candidates we filtered elements that could already have been detected with the regular expression method described above (see Section 4.1.). The filtering in Tables 5 and 7 is based on the manual annotation. Additionally, we used knowledge about previous disfluencies to add features to the feature set:

²<http://www.cs.waikato.ac.nz/ml/weka/>

- previous disfluency in segment (yes/no)
- distance to previous disfluency (0 in case there is none)
- distance to previous disfluencyTYPE (is either NLFP , ABW or REPET)

type	accuracy	prec	rec	f
non oversampled				
disfluent	89.7	80.7	58.4	67.7
non-disfluent		91.1	96.8	93.9
oversampled				
disfluent	80.5	54.3	60.8	57.4
non-disfluent		88.9	86.0	87.4

Table 5: Binary classification with filtering

Table 5 shows the results of the binary classification after NLFP , ABW and REPET have been filtered out. The overall accuracy rate increased and the classification of both non-disfluent and disfluent items increased. Again using oversampling did not improve the results, but rather decreased results on the overall accuracy and classification.

The next step aims at a full classification of all six disfluency types.

disfl class	accuracy	prec	rec	f
non-oversampled				
NLFP	86.4	55.5	45.5	50.0
LFP		64.3	51.4	57.1
abutt		29.8	4.5	7.8
abw		67.3	79.6	72.9
repai		45.2	12.6	19.7
repet		64.7	50.0	56.4
none		89.8	97.3	93.2
oversampled				
NLFP	78.4	54.4	53.4	53.9
LFP		38.5	54.4	45.1
abutt		14.6	21.1	17.2
abw		68.9	61.1	64.8
repai		25.8	24.0	24.8
repet		51.4	57.7	54.4
none		91.7	87.7	90.1

Table 6: Full classification, no filtering

Table 6 shows results on all disfluency types without any previous knowledge. What can be seen from these results is that NLFP , REPET and ABW do not benefit from machine learning methods. They are more reliably classified by the method presented above (see Section 4.1.). As the results for the other three classes (LFP , REPAI and ABUTT) are low, we also oversampled the data, so that all six classes match in distribution. But oversampling increased the overall error rates, as in the previous experiments. The classification of LFP decreased significantly, whereas the classification of ABUTT and REPAI increased. This indicates that oversampling can be beneficial in some cases, but not in general.

Table 7 shows results for the final experiment on classifying LFP , ABUTT and REPAI after filtering for both non-disfluent items and for NLFP , ABW and REPET. As the results show there is still a considerable error rate, but the

disfl class	accuracy	prec	rec	f
non oversampled				
LFP	82.1	83.4	91.1	87.1
abutt		76.2	73.0	74.6
repai		84.3	77.0	80.5

Table 7: Full classification, binary and regular expression based filtering

classification of single items reaches similar values as the classification of the classes categorized by the regular expression based method (see Table 3). Again oversampling did not improve the results, but rather decreased them.

A closer look at the feature ranks revealed that the most important information was provided by the POS tag of the item to be classified and the POS tags of the surrounding items. Additionally, the length of the segment in which the item occurred was of importance. Some information was gained by looking at the distance to the disfluency start and the average length of the segments uttered by the speaker. Very little information was retrieved from information about the distance of the current item to the previous disfluency type (REPET , NLFP and ABW), whereas the general distance to some previous disfluency was more helpful. Contrary to what has been proposed in the literature and to our own expectation information about gender gave little information.

This is also supported by rules that could be derived from the learned decision tree. One example is presented in Figure 3.

The rules in Figure 3 Example 1 state that in case a certain combination of tags occurs (IN INP IN INP) and that the disfluency started with one of these items, the item under consideration is very likely an abandoned utterance (ABUTT). But, if the disfluency did not start within these items and the speaker has so far uttered more than 48 segments and the gender is female it is very likely a lexicalized filled pause (LFP), but if the gender is male and the average length of what this person says exceeds a certain number (7) the item is also very likely a abandoned utterance (ABUTT). This shows that the gender information comes in very late. The most informative are the POS tag information, distance to the disfluency and talkativity information on the current speaker.

Some shorter rules are presented in Figure 3 Example 2, which state that if the current tag is marked as UH combined with a coordinating conjunction in a rather short segment (≤ 11), where a disfluency occurs the item is an abandoned utterance (ABUTT). In case there is no further disfluency, it results in a lexicalized filled pause (LFP).

Another rule presented in Figure 3 Example 3 indicates a repair (REPAI) is that if the token is tagged as interpunctuation, which is preceded by a coordinating conjunction and the position of the item is at the beginning of the segment and the segment exceeds a certain length, the item belongs to a repair.

4.3. Comparison with Zechner

This final section contains an evaluation on holdout data with the three stage classifier, using regular expression

```

Example 1:

segmentLength <= 11 & tag = INP & lprevTag = IN & 2nextTag = INP & lnextTag = IN &
distanceToDisflStart <= 1 --> ABUTT

distanceToDisflStart > 1 & distanceToDisflStart <= 3 &
segmentsSF <= 48 --> ABUTT

segmentsSF > 48 & gender = f --> LFP

gender = m & averageSegment <= 7 --> LFP
averageSegment > 7 --> ABUTT

=====

Example 2:

segmentLength <= 11 & tag = UH & lprevTag = CC &
previousDisfl = yes --> ABUTT

previousDisfl = no --> LFP

=====

Example 3:

segmentLength <= 11 & tag = INP & lprevTag = CC & segmentPos <= 1 &
segmentLength <= 5 --> ABUTT

segmentLength > 5 --> REPAI

```

Figure 3: Example for rules from learned decision tree.

based detection, binary classification through a trained model and the classification of the remaining disfluency types also based on a trained model.

Table 8 shows the *f*-measure results on the different disfluency classes obtained by the three stage classifier presented here compared to the classifier presented by Zechner (2001). As can be seen the detection of NLFP and REPET is considerably better than the results reported in (Zechner, 2001). False starts are divided into two categories: full sentences and non-finished sentences. These categories are best compared to ABW and ABUTT. The detection of ABW is slightly better than the detection of false starts. The classification of ABUTT is considerably worse. At least for the non-oversampled case.

If the two abandoned types (ABUTT and ABW are conflated, a *f*-measure of 0.61 is achieved, which is better than the false start on non-finished sentences reported in (Zechner, 2001), but worse than the false starts on full sentences. If the two repair types (REPET and REPAI) are conflated, a *f*-measure of 0.49 is achieved, which is better than the *f*-measure of 0.41 reported by Zechner (2001). The overall accuracy is 97.21%. A number of tokens were detected as disfluent, but were classified as the wrong type. These account for 1.21% of cases. These two conflations are sound based on two reasons: First, they are easily confused as an abandoned word can be part of an abandoned utterance and a repetition can be part of a repair. Second, the compar-

ison with Zechner (2001) is easier, as there is not such a fine-grained distinction between these two types.

The results using a model built on oversampled data does not improve the results on LFP and REPAI. Only ABUTT improves, but does not reach results reported by Zechner (2001). The same is true when classes are conflated as has been done with the non-oversampled data above.

5. Conclusions

We presented a three-stage procedure to detect and classify disfluencies in multi party dialogues. We could show that our method outperforms another system that also captures disfluencies in multi party dialogues.

The main differences are that we used a more fine-grained distinction for various disfluency types and the three stages built explicitly on each other. In Zechner (2001) previously detected disfluencies were removed, but information on them was not used as a feature in the learning system. The finer grained distinction could account for the lower results achieved by our system.

Another issue that was touched in this work was observations from descriptive work on disfluencies. The features in our learning system were based on such observations (e.g. gender information, talkativity information, etc.). But in the rules and the feature ranks we could only find some support for these observations (talkativity seems to matter, but gender does not). Also the position of the item to be

Class (Z)	Class	Result (Z)	Result (NO)	Result (O)
NLFP	NLFP	0.45	0.95	0.95
Discourse Markers	LFP	0.30	0.41	0.23
Repairs	REPET	0.41	0.99	0.99
Repairs	REPAI	0.41	0.14	0.10
False Start	ABW	0.56/0.94	0.97	0.97
False Start	ABUTT	0.56/0.94	0.015	0.07

Table 8: Result on the three stage classifier of disfluencies (non-oversampled (NO) and oversampled (O)) compared to results reported in Zechner (Class(Z) and Result (Z)).

classified did not play a major role. This means either that this information is not discriminative enough or that there were too few examples to allow for better discrimination. Additionally, the observations were made on two-party dialogues, some of them human-human dialogues and others human-machine dialogues. A more detailed evaluation of the disfluencies in multi party dialogues could be worthwhile to undertake in order to find differences between different types of data. It would also be interesting to use real speech recognition output and features that are also based on the speech data (pauses, prosodic information).

Acknowledgments. This work has been supported by the DFG under grant STR 545/2-1,2 within the DIANA-Summ project and by the Klaus Tschira Foundation.

References

- Baron, Don, Elizabeth Shriberg & Andreas Stolcke (2002). Automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues. In *Proceedings of the International Conference on Spoken Language Processing*, Vol. 2, pp. 949–952. Denver, USA.
- Brill, Eric (1994). Some advances in transformation based part-of-speech tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence*, Seattle, Wash., 1–4 August 1994, pp. 722–727.
- Cohen, Jacob (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46.
- Core, Mark G. & Lenhart K. Schubert (1999). A syntactic framework for speech repairs and other disruptions. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, College Park, Md., 20–26 June 1999, pp. 413–420.
- Flaiss, Joseph L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.
- Godfrey, John J., Edward Holliman & J. McDaniel (1992). Switchboard: Telephone speech corpus for research and development. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, San Francisco, Cal., USA, pp. 517–520.
- Heeman, Peter A. & James F. Allen (1999). Speech repairs, intonational phrases, and discourse markers: Modeling speakers’ utterances in spoken dialogue. *Computational Linguistics*, 25(4):527–571.
- Janin, Adam, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke & Chuck Wooters (2003). The ICSI meeting corpus. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, China, 6–10 April 2003, pp. 364–367.
- Johnson, Mark & Eugene Charniak (2004). A tag-based noisy channel model of speech repairs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 21–26 July 2004, pp. 33–39.
- Lendvai, Piroška, Antal van den Bosch & Emiel Kraemer (2003). Memory-based disfluency chunking. In *Proceedings of Disfluency in Spontaneous Speech Workshop* Göteborg University, Sweden, September 5-8, 2003, pp. 63–66.
- Lickley, Robin J. (2001). Dialogue moves and disfluency rates. In *ITRW on Disfluency in Spontaneous Speech*, Edinburgh, Scotland, UK, August 29-31, 2001, pp. 93–96.
- Liu, Yang, Elizabeth Shriberg, Andreas Stolcke & Mary Harper (2005). Comparing HMM, maximum entropy, and conditional random fields for disfluency detection. In *Proceedings of the 9th European Conference on Speech Communication and Technology (INTERSPEECH ’05)*, Lisboa, Portugal, 4–8 September 2005.
- Nakatani, Christine & Julia Hirschberg (1993). A speech-first model for repair detection and correction. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, 22–26 June 1993, pp. 46–53.
- Pakhomov, Sergey & Guergana Savova (1999). Filled pause distribution and modeling in quasi-spontaneous speech. In *Proceedings of the International Conference of Phonetic Sciences*, San Francisco, USA, August 1999.
- Popescu-Belis, Andrei, Alexander Clark, Maria Georgescu, Denis Lalanne & Sandrine Zufferey (2004). Shallow discourse processing using machine learning algorithms (or not). In S. Bengio & H. Bourlard (Eds.), *Machine Learning for Multimodal Interaction*, Vol. 3361, Springer Lecture Series in Computer Science, pp. 277–290. Springer.
- Shriberg, Elizabeth Ellen (1994). *Preliminaries to a Theory of Speech Disfluencies*, (Ph.D. thesis). University of California at Berkeley.
- Siegel, Sidney & N. John Castellan (1988). *Nonparametric Statistics for the Behavioral Sciences* (2nd ed.). New York: McGraw-Hill.
- Snover, Matthew, Bonnie Dorr & Richard Schwartz (2004). A lexically-driven algorithm for disfluency detection. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Boston, Mass., 2–7 May, 2004, pp. 157–160.
- Stouten, Frederik & Jean-Pierre Martens (2004). Coping with disfluencies in spontaneous speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*, Vol. 2, pp. 1513–1516. Jeju Island Korea.
- Teufel, Simone & Marc Moens (2002). Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.
- Zechner, Klaus (2001). *Automatic Summarization of Spoken Dialogues in Unrestricted Domains*, (Ph.D. thesis). Language Technology Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA.