Available online at www.sciencedirect.com
ScienceDirect
Speech Communication 48 (2006) 1590–1606
SPEECH COMMUNICATION
www.elsevier.com/locate/specom

Coping with disfluencies in spontaneous speech recognition: Acoustic detection and linguistic context manipulation

Frederik Stouten [a,*], Jacques Duchateau [b], Jean-Pierre Martens [a], Patrick Wambacq [b]

[a] *ELIS, University of Ghent, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium*
[b] *ESAT, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Heverlee, Belgium*

Received 25 March 2005; received in revised form 14 April 2006; accepted 26 April 2006

Abstract

Nowadays read speech recognition already works pretty well, but the recognition of spontaneous speech is much more problematic. There are plenty of reasons for this, and we hypothesize that one of them is the regular occurrence of disfluencies in spontaneous speech. Disfluencies disrupt the normal course of the sentence and when for instance word interruptions are concerned, they also give rise to word-like speech elements which have no representation in the lexicon of the recognizer. In this paper we propose novel methods that aim at coping with the problems induced by three types of disfluencies, namely filled pauses, repeated words and sentence restarts. Our experiments show that especially the proposed methods for filled pause handling offer a moderate but statistically significant improvement over the more traditional techniques previously presented in the literature.

Keywords: Disfluency handling; Spontaneous speech recognition; Disfluency detection

1. Introduction

The automatic recognition of spontaneous speech is currently a hot topic. Practical applications of spontaneous speech recognition include voice operated telephone services, automatic closed captioning for TV programs, automatic transcription of meetings, etc. Yet, the recognition accuracy of freely spoken language is quite poor when compared to that of dictated speech: while the state-of-the-art word error rates (WER) for large vocabulary speaker-independent dictation and broadcast news transcription are of the order of 5% (Stouten et al., 2003) and 15% (Beyerlein et al., 1999; Gauvain et al., 1999) respectively, the WER for the transcription of meetings (Yu et al., 2000) can be as large as 40%.

One important reason for this deficiency of spontaneous speech recognizers is the lack of a good language model built on a large amount of

[*] Corresponding author.
E-mail addresses: fstouten@elis.ugent.be (F. Stouten), Jacques.Duchateau@esat.kuleuven.be (J. Duchateau), martens@elis.ugent.be (J.-P. Martens), Patrick.Wambacq@esat.kuleuven.be (P. Wambacq).

0167-6393/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2006.04.004

spontaneous speech transcripts. While typical stochastic language models for read speech recognition rely on vast amounts of training material (Adda et al., 1999), no comparable amounts of written transcripts of casual language are available.

On top of that, the occurrences of disfluencies in casual speech may further complicate the estimation of a robust spontaneous language model. In the literature different approaches to spontaneous language modeling have already been pursued. In (Ma et al., 2000), one tries to incorporate knowledge from discourse theory, and one argues that sentences typically start with given information and end with new information, and disfluencies mostly occur in the given information part of the sentence. By applying *specialized* language models for the two sentence parts, one obtains a marginal drop of the WER (0.3% absolute) on the recognition of spontaneous telephone conversations from the Switchboard corpus (see Godfrey et al., 1992 for more information on this corpus).

In (Zechner and Waibel, 1998) the potential of *N*-best list rescoring on the basis of information from a chunk parser has been explored. The underlying assumption is that the chunker bears information that can help to discriminate between syntactically acceptable and syntactically anomalous hypotheses. It was demonstrated that this technique yields a marginal drop of the WER (0.3% absolute) on Switchboard.

Even if one disposes of a language model comprising context-dependent probabilities for disfluencies, it may be beneficial to remove the disfluency from the context when predicting *regular words* (as opposed to disfluencies) occurring right after that disfluency. Stolcke and Shriberg (1996) have already investigated this technique, but their experiments on Switchboard did not show any significant performance gain. In Section 3.1 of this paper we propose a more flexible manipulation of the prediction context. In that approach, regular words are predicted with and without the disfluencies being removed from the context. These predictions are then allowed to compete with each other.

On the acoustical–lexical front, the literature also describes several solutions to disfluencies in recognition systems. For instance in (Schramm et al., 2003) a data-driven lexical modeling technique was applied to construct a lexical model with many pronunciation variants for a filled pause (FP). By substituting the single-pronunciation FP model from the baseline system by this new more complex model, a 2% absolute (=7.8% relative) reduction of the WER on a highly spontaneous medical transcription task was achieved.

A draw-back of the previously proposed techniques for disfluency handling is their assumption that the decoder part of the recognizer is capable of producing reliable disfluency hypotheses. We argue that in particular for filled pauses this assumption is often wrong. In fact, we will demonstrate that by introducing a specialized FP detector operating independently of the decoder, and by supplying the output of that detector to the decoder, it is possible to achieve a more significant improvement of the recognition accuracy with only a very limited increase of the computational load.

The paper is organized as follows. In Section 2 we present some background information concerning the prevalence of different disfluency types in existing spontaneous speech corpora and we discuss how these disfluencies can affect the recognition of spontaneous speech. In Section 3 we discuss the novel techniques we propose to remedy these problems. From the viewpoint of the decoder these techniques can be divided into internally and externally informed techniques. An internally informed technique relies on FP hypotheses that are being generated by the decoder, whereas an externally informed technique is helped by a disfluency detector working independently of the decoder, as a kind of extra acoustic front-end. In Section 4 we propose such a novel detector. In Section 5 we report recognition results that were obtained with the newly proposed disfluency handling techniques, either when applied individually or in combination with each other. In Section 6 we discuss some additional analyses we performed in order to assess in detail the behavior of our disfluency handling methods and their effectiveness with respect to maximum performance gains we were able to estimate from experimental data.

2. Disfluencies in spontaneous speech

In this section we assess the prevalences of three important disfluency types in spontaneous speech, and we briefly discuss the major effects these disfluencies can have on the automatic speech recognition process. As mentioned before, we make a distinction between disfluencies and regular words and we express the prevalence of a particular disfluency type by means of its *disfluency rate*. The latter is defined as the number of disfluencies of that type divided by the total number of word tokens.

2.1. Speech corpora

In order to gain some insight in the language dependency of disfluency prevalences, we have performed measurements on two distinct corpora.

(1) The first corpus is an excerpt from the Spoken Dutch Corpus (CGN) (Goedertier et al., 2000; Martens et al., 2002). This excerpt represents about 12 h of recordings containing spontaneous speech of 130 Flemish speakers. All together these recordings contain nearly 130K word tokens. Orthographic transcripts and manually verified word-level segmentations are available for all this material. The corpus is very diverse: it comprises spontaneous interviews, broadcast field reports, political debates and panel discussions. For reasons that will become clear later the corpus was further divided into a *training corpus* (91 files and about 11 h of speech) and a *test corpus* (16 files and about 1 h of speech). The statistical analysis presented in this section will be performed on the training corpus. Since the test corpus will be used to assess the effects of our disfluency handling methods on the speech recognition performance, we have made sure that the 27 speakers appearing in the test corpus did not appear in the training corpus.

(2) The second corpus is the Switchboard-1 corpus. From this corpus we held out the 2001 HUB5 benchmark set for statistical analysis (this section) and for recognition system evaluation (Section 5). It is composed of recordings of 20 informal telephone conversations (=40 speakers) in American English (Godfrey et al., 1992). The recording time is about 2 h and the recordings contain about 20K word tokens. The orthographic transcripts of the material can be found on the Web: *ftp://jaguar.ncsl.nist.gov/lvcsr/mar2001*.

We have investigated three types of disfluencies, namely filled pauses (FPs), word repetitions (WRs) and sentence restarts (SRs). According to (Shriberg, 1996), these three types represent about 85% of all the disfluencies occurring in the Switchboard corpus. In the following sections we briefly describe the measurements we made and the results they produced.

2.2. Filled pauses (FPs)

It is commonly acknowledged that FPs are the most frequently occurring disfluencies in spontaneous speech. A filled pause usually appears as an interjection, like "uh" or "uhm", but the acoustic properties of the interjections may be language dependent. Consider two examples which have been extracted from the CGN and translated to English:

(1) oh I read all the *uh* books of Simenon but I *uh*.
(2) *uh* particularly *uhm* Dutch literature *uhm*.

They illustrate that fillers can occur at many positions in the utterance.

Counting the number of FPs in the two corpora was rather easy because both the CGN and the Switchboard orthographic transcription protocols instructed the transcribers to stick to a restricted list of fillers to encode an FP.

The mean FP rate in the CGN dataset was equal to 2.7%. However, a number of speakers had an FP rate of more than 10%. In the Switchboard dataset the measured FP rate was equal to 3%. This rate is much larger than the 1.7% reported by Shriberg (1996) for another subset of the Switchboard corpus.

Of all 596 FPs observed in Switchboard, 106 were sentence initial, 111 occurred at the end of a sentence, and 46 others were actually isolated sentences, meaning that they contained no other words. These figures imply that more than 50% of the FPs (333 in total) were sentence medial FPs. In the CGN dataset on the other hand, we found that 387 of the 445 FPs were sentence medial. This discrepancy may originate from the different types of speech material in the two corpora.

2.3. Word repetitions (WRs)

One strategy for a speaker to gain some time to think is to repeat a word once or several times before continuing with the rest of the sentence. If this happens, the last word of the repeated word sequence is considered as the regular word whereas the others are designated as disfluencies. The following two examples were selected from the CGN and translated to English (the WRs are put in italic):

(1) *I I* also work *with with* music. I work *I I I* mean dance music.
(2) well *what what* are children supposed to do?

Counting word repetitions in a corpus is seemingly very simple: search for repetitions of the same word and count the number of disfluencies they represent. In practice, it is much more complicated than that. For instance, if an interjection occurs between identical words, there is still a word repetition involved. This complication is easy to accommodate. A more problematic complication is the presence in most languages, including English and Dutch, of grammatically correct word repetitions. Consider the following two English examples:

(1) We think *that that* man at the station was drunk.
(2) I still have to read *many many* articles on this topic.

None of the highlighted word repetitions should be counted as a disfluency, but we only know this because we understand the meaning of these sentences. Since it is currently impossible to make a simple and reliable semantic parser of spontaneous speech, we have chosen for a semi-automatic procedure to count the WR type disfluencies. In a first pass, we create a list of all sentences containing the following events: two identical words in a row, or two identical words separated by a filled pause. These events are then marked by a human expert as a word repetition or a grammatically correct word sequence.

The WR-rate in the CGN dataset turned out to be about 0.9%. In the Switchboard set it was about 1.4% and equal to the percentage also reported by Shriberg (1996) for another dataset. In the Dutch data, there were less than 0.07% word repetitions that consisted of three or more words. In the American data, this figure was much larger and close to 0.4%. Both datasets show very few FP interjections between repeated words. This suggests that WRs and FPs are two clearly distinct speaker strategies for gaining time.

Since for the CGN data we also had manually verified word-level segmentations at our disposal, and since these segmentations also reveal inter-word pauses of more than 100 ms long, we were able to determine how many times repeated words are being separated by a (silent) pause. This happened in about 40% of the cases.

Another interesting finding was that the top-20 of repeated words in the CGN are all monosyllabic *function* words like "en", "een", "dat"... Moreover, this top-20 can explain 78% of all the WRs encountered in the corpus.

2.4. Sentence restarts (SRs)

A sentence restart (SR) is defined here as a situation in which the speaker makes the initial part of a started sentence obsolete by the succeeding words. The following examples represent instances of such SRs that were found in the CGN and translated to English (the obsolete part, also called the reparandum, is put in italics):

(1) *is uh* did Agalev abandon you?
(2) *in a situation with uh* in a country with two speed levels.

Obviously, sentence restarts cannot be retrieved automatically from the orthographic transcripts, unless they are explicitly annotated as such in these transcripts. Since no such annotations were available neither for CGN nor for Switchboard, we had to retrieve the SR-rate by means of a manual inspection of the transcripts. We have only performed this on the Switchboard dataset. For this set we found 112 restarts corresponding to an SR-rate of about 0.5%. This rate was obtained as the number of sentence restarts (one per SR) divided by the total number of words. We also found that about 30% of the reparandi (33 instances) ended on a filled pause. Recalling that there were only 333 sentence medial FPs, this means that about 10% of all these FPs actually initiate a sentence restart.

2.5. Main effects on the automatic recognition process

One effect of an FP is that it introduces a new word (denoted as "uh") that is normally not included in the lexicon of a read speech recognizer. Obviously this effect can easily be accounted for by adding this word to the lexicon. One option is to add it with a pronunciation "uh" representing a dedicated whole-word speech unit whose acoustic model is trained on FP utterances. Another option is to add it with a pronunciation model specifying all the likely pronunciations of the FP in terms of the regular phonemes of the language.

An effect that is common to all types of disfluencies is that they disrupt the normal word flow. This disruption implies that the spoken word sequence no longer matches well with a language model (LM) that was retrieved from text material not containing any disfluencies. Consequently, the LM probability of the correct word sequence may comprise a number of low backoff probabilities, and the decoder may therefore be inclined to select a more likely sequence by assigning the FP interval to a short function word (e.g. "a", "the", etc.) that is acoustically similar to the FP, or to a syllable of a content word, a syllable that acoustically sounds like an FP. In both cases the decoder will produce wrong word hypotheses which will on their turn affect the word prediction capability of the LM in the vicinity of the disfluency. Consequently, it can be anticipated that one disfluency can be responsible for more than one error in the recognition output. In their paper Adda-Decker et al. (2003) report a figure of about 1.5 errors per disfluency for a French spontaneous corpus. If this figure would generalize to our speech data it would mean that a disfluency rate of 3–5% could be responsible for a WER contribution of 4.5–7.5% absolute. In Section 6 of this paper we describe an experiment to assess the expected number of errors per filled pause on our own material from the Spoken Dutch corpus. We also found a figure of 1.5 word errors per FP. By also taking the average WER of our recognizer into account, we were able to show that the number of errors per FP one can hope to correct by introducing FP handling methods is of the order of 0.7 words per FP. That would then correspond to an absolute decrease of the WER by 2.1–3.5.

3. Proposed methods for coping with disfluencies

In this section we propose a number of so-called *internally informed* and *externally informed* search strategies for coping with disfluencies in spontaneous speech recognition.

In an internally informed search strategy, the acoustic models, the lexicon and the LM are all together responsible for hypothesizing disfluencies and the search engine must be adapted to undertake special actions when such hypotheses are generated.

An externally informed search strategy uses an external disfluency detector to spot the disfluencies. We argue that such a strategy has a lot of potential in the case of FPs. First of all it is anticipated that FPs have some well-defined acoustic and prosodic properties (Batliner et al., 1995; Quimbo et al., 1998; Gabrea and O'Shaugnessy, 2000; O'Shaugnessy, 1993). Secondly, the acoustic models of a speech recognizer are usually blind for prosody, and as such they may be unable to produce enough evidence for distinguishing an FP from a function word like *a* (English) or *de* (Dutch), or from the initial syllable of a content word like *a̲bove* (English) or *g̲etal* (Dutch).

In what follows we propose three novel internally informed search strategies for dealing with FPs, WRs and SRs, as well as two externally informed search strategies for coping with FPs.

3.1. Internally informed strategies for coping with disfluencies

As already stated in Section 1, one of the hypotheses explaining the difficulty of modeling spontaneous language by means of N-grams points explicitly to disfluencies: as N-grams base their word prediction on a local context of $N-1$ previous words, intervening disfluencies render this context less uniform. Or put differently, the prediction of the next word would be more accurate if it were based on a context from which the disfluency was removed. Obviously, removing disfluencies (one or more in a row) also implies that the word context to take into account in the decoder is extended to the left with regular words appearing in front of these disfluencies.

Consider for instance the example "this is what *uhm* I think", and presume that the LM is a trigram model. We argue that in this case the word "I" would be better predicted by the context "is what" than by the context "what uhm". Nevertheless, Stolcke and Shriberg (1996) came to the surprising conclusion that discarding FPs from the trigram context actually increases the perplexity. However, they were looking at speech stretches that were isolated on the basis of acoustic criteria (the presence of large silent pauses), meaning that the FPs occurring at sentence boundaries often appeared in the middle of such a stretch. By only discarding sentence internal FPs the perplexity did decrease indeed. In the material of (Peters, 2003) the speech stretches all corresponded to sentences and therefore all FPs were sentence internal. For this material, the discarding strategy resulted in a 4% decrease of the overall perplexity and a 30% decrease of the perplexity of the first word after the FP.

In (Siu and Ostendorf, 1996) and (Shriberg and Stolcke, 1996), it is nevertheless shown that in some cases FPs *are* good predictors for the following words: they often tend to precede a less frequently used word. Therefore, simply discarding the FPs from the context is perhaps not always the best solution. This conclusion also holds for repeated words which are part of a grammatically correct word sequence, like in the example "I hope that that work is at least done properly now".

In an attempt to account for the above observations we propose a context manipulation that provides two or more options and that leaves it to the recognition system to select the most likely option on the basis of all its knowledge (acoustic, lexical and linguistic). We have developed three models: one model to apply in case a word repetition is hypothesized, and two models to apply in case a filled pause is hypothesized.

3.1.1. The repetition model

The model for handling word repetitions is sketched in Fig. 1. It presumes that the LM is a trigram model. The upper path illustrates the normal LM procedure. If the hypothesized word *B* appears to be a repetition of the previous word, then the prediction of the next word *C* is based on the context *B B*. The lower path represents the alternative of predicting the word *C* on the basis of *A B*, the context of which is obtained by simply ignoring the repeated word *B*.

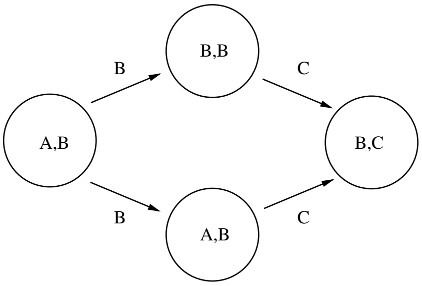

Fig. 1. The model for repetitions.

3.1.2. The hesitation model

The hesitation model is activated in case a filled pause is detected. The model is depicted in Fig. 2 with "uh" denoting an FP. The model proposes two alternatives to the search engine: (1) the standard solution (upper path) in which the filled pause is kept in the context for predicting the subsequent words, and (2) an alternative solution (lower path) in which the filled pause is removed from that context.

3.1.3. The restart model

Starting from the observation that many filled pauses announce a sentence restart, we have conceived a restart model (Fig. 3) that is activated everytime an FP is hypothesized by the decoder. The lower path models the fact that an FP causes a reset of the language model: the left context is reset to the sentence start symbol $\langle S \rangle$. It is clear that a sentence restart can also occur after a regular word, but a pilot experiment described in (Duchateau et al., 2003) demonstrated that activating the restart model for each word hypothesis causes an over-generation of restart hypotheses, and has a negative impact on the recognition accuracy.

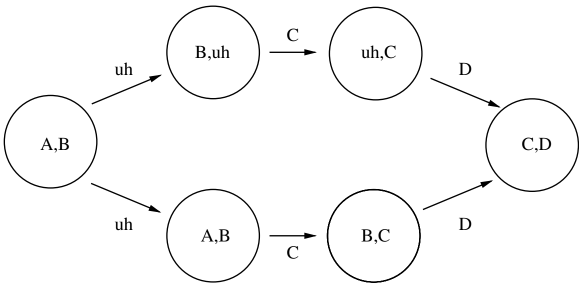

Fig. 2. The model for hesitations.

When successful, the hesitation and the restart model can obviously be combined into one more complex model to be applied whenever an "uh" is hypothesized.

3.2. Externally informed strategies for coping with FPs

In this section, we propose two externally informed strategies for coping with filled pauses. They rely on the outputs of an external FP detector which works independently of the decoder part of the recognizer (see Section 4). In the presented work, the FP detector produces variable length FP segments, and each segment has an associated posterior probability $P(\mathrm{FP}|\boldsymbol{X})$. The symbol $\boldsymbol{X}$ stands

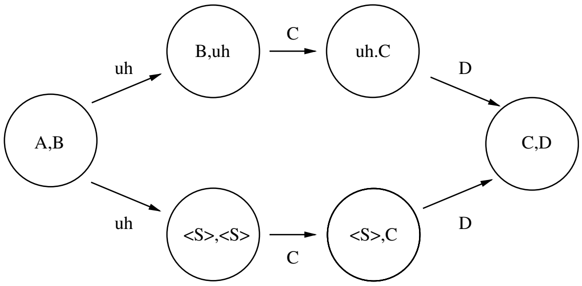

Fig. 3. The model for sentence restarts.

for the acoustic observations in and around the hypothesized FP segment and the posterior probability is hereafter called the FP score.

3.2.1. Frame dropping

If the FP score of a segment is high, one can expect that all the frames of that segment are FP frames. The idea of frame dropping is to let the decoder discard these frames. An advantage of the technique is that it can easily be integrated in any speech recognition system. All it takes is not to supply the FP frames to the decoder. Another advantage is that it can also be applied in combination with a decoder that does not even incorporate an FP model in its lexicon.

Obviously, frame dropping is a pretty drastic method which is bound to deteriorate the recognition performance if too many regular speech frames would be discarded. For instance, it can happen that the speaker starts by saying the word “the” and continues by prolonging the word, more or less gradually shifting to a filled pause. The correct handling of such a filled pause is problematic, because the FP detector is bound to indicate all the vocalic frames of the fragment as constituting an FP, and discarding all these frames may prevent the decoder from hypothesizing the word “the” as it should. Anyway, we expect that frame dropping will work best in combination with an FP detector that does not often produce large FP scores for non-FP (NFP) segments. This means an FP detector with a high precision.

3.2.2. LM adaptation

Because of the potential danger of frame dropping we have also conceived a second strategy which is less categorical in its interpretation of the FP character of the frames. In this so-called LM adaptation strategy, the normal LM probability of a word hypothesized in a time interval (t_1, t_2) which overlaps with an FP segment emerging from the external FP detector, will be replaced by a new probability that depends on (1) the identity of the word hypothesis, (2) the distance between t_1 and the FP segment start, (3) the fraction of (t_1, t_2) falling into this FP segment (the overlap fraction), and (4) the value of the FP score that was computed for this FP segment.

The LM adaptation procedure is activated every time a word hypothesis is generated. It works as follows (see also Fig. 4):

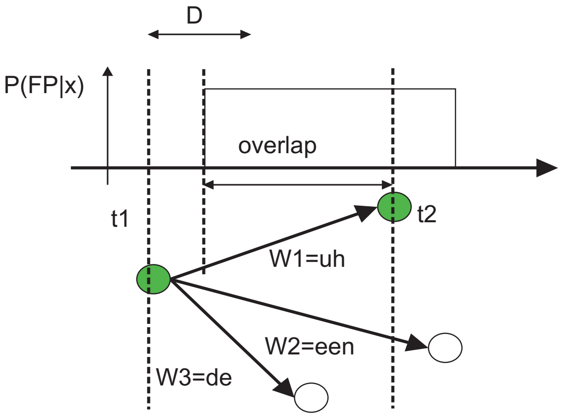

Fig. 4. LM adaptation is examined for word hypotheses (bottom) which exhibit some overlap with FP segments (top) produced by the FP detector.

(1) If the hypothesized word starts at a time t_1 which is further than some threshold D away from the start of an FP segment detected by the external FP detector, then leave the LM probability unaltered, else continue with the next step.
(2) If the overlap fraction between the hypothesized word and the FP segment is smaller than 50%, then take no special action either, else continue with the next step.
(3) If the hypothesized word is “uh”, then replace the normal LM score by a predefined value C_1. If the hypothesized word is not “uh”, then subtract some predefined amount C_2 from the normal LM score which is log $P(\text{word}|\text{context})$.

By manipulating C_1, C_2 and D it is possible to control the impact the FP detector can have on the recognition output. An alternative for the probability substitution outlined in point (3) would be to replace log $P(\text{“uh”}|\text{context})$ by the logarithm of the FP score emerging from the FP detector. We did test this approach, but it did not outperform the simpler and easier to control strategy outlined in point (3).

It is clear that the LM adaptation strategy uses all the available knowledge sources to make a distinction between the true and false FP segments proposed by the FP detector. Therefore LM adaptation could well be able to identify the false alarms emerging from the external FP detector. Consequently, we expect the technique to be most effective if it is applied on almost all the FP segments appearing in the speech. This means when it is combined with an FP detector with a high recall.

4. An independent detector of filled pauses

We argue that the externally informed methods for coping with FPs will have an advantage over the internally informed strategies if they can rely on an FP detector that can spot the FP segments with a much higher accuracy than the decoder of the speech recognizer would be able to do. Therefore, we have conceived an FP detector which will base its decisions on MFCC vectors and on additional acoustic and prosodic cues that are not available to the acoustic models of the decoder (Stouten and Martens 2003). The proposed detector first performs a blind segmentation of the speech into silent and phoneme-like segments. Then it classifies the non-silent segments as FP or NFP segments. We conjecture that the segmental framework facilitates the introduction of prosodic cues related to pitch and duration in the classification process. Such cues are invisible to the acoustic models embedded in the speech recognizer.

Next, we will describe the segmentation of the speech, the extraction of appropriate features to represent the segments and the classification of these segments into FP and NFP segments on the basis of these features. The feature selection and the training and evaluation of the FP detector are all achieved on the basis of the previously described CGN training and test corpora. We do believe however that the methods and results reviewed here are also relevant for the construction of e.g. an American English FP detector.

4.1. Speech segmentation

In order to create a segmental description of the speech, we first construct a cepstral difference pattern, and we then extract potential segment boundaries at the maxima in this pattern.

Starting from the standard MFCC vectors $c(t)$ (t is the frame index) extracted by the front-end of the recognizer, we derive a first cepstral difference $d_c(t)$ according to

$$d_c^2(t) = \sum_{i=0}^{12} \left[\frac{\sum_{j=1}^{N_w} j[c_i(t+j) - c_i(t-j)]}{2\sum_{j=1}^{N_w} j^2} \right]^2 \qquad (1)$$

Since $c_0(t), \ldots, c_{12}(t)$ represent the elements of $c(t)$, the computed $d_c(t)$, which is always positive, represents the norm of the slope of the best linear regression of the cepstral vectors in a window of $2N_w + 1$ frames centered around the time of interest t. The pattern $d_c(t)$ is then further smoothed by means of a three point FIR filter to

$$d(t) = \frac{1}{4}d_c(t-1) + \frac{1}{2}d_c(t) + \frac{1}{4}d_c(t+1) \qquad (2)$$

and from this pattern the segment boundaries are derived. In order to do so, a robust left-to-right minimax algorithm (Vorstermans et al., 1996) tracks the locations of clear maxima in $d(t)$. A clear maximum is defined as one which is considerably higher than the largest of the two minima surrounding this maximum. A silence detector is also integrated in the segmentation algorithm. It detects intervals of at least three successive frames having a $c_0(t)$ (called the log-energy) that is not more than 3 dB above an adaptive log-energy noise floor, computed on the basis of minimum statistics. If a minimum in $\max[c_0(t-1), c_0(t), c_0(t+1)]$ is encountered that is more than 3 dB below the noise floor, the noise floor will be replaced by this minimum. If the minimum is higher than the threshold, but lower than the lowest minimum since the last threshold adaptation, its value and position will be stored as a potential new minimum. If no silence was detected in a period of a minimal length (e.g. 2 s), the energy threshold is adapted to the value of the lowest potential minimum and the search for silences will be resumed from there.

4.2. Feature identification

An appropriate acoustic and prosodic feature description for the created non-silent segments has to be conceived now. To that end we have performed a statistical analysis of our CGN training corpus which contains 3255 FP intervals, 75% of which are longer than 0.2 s and 87% longer than 0.15 s. In the rest of our research we have considered only those FPs which are longer than 0.15 s as genuine FPs.

If more than 50% of the frames of a segment emerging from the previously described blind segmentation fall into such a genuine FP interval, the reference label of that segment is FP, otherwise it is NFP. By comparing the cumulative distribution functions (CDFs) of a feature for the FP and the NFP segments, we can identify features that are expected to contribute to the discrimination between FP and NFP segments. For figures showing cumulative distributions of the individual features, the reader is referred to Stouten and Martens (2003). In the present paper we only briefly review the features that were investigated and the discriminative power these features seem to have in the CGN training data.

4.2.1. Segment duration

The first feature we have investigated is segment duration. Our measurements showed that FP segments tend to be longer than NFP segments. This result confirms the observations also made by e.g. (Gabrea and O'Shaugnessy, 2000). The FP and NFP segment durations both seem to exhibit Gamma distributions, but with clearly different parameters. The mean FP length is about 0.25 s (SD = 0.15 s), the mean NFP length is only 0.11 s (SD = 0.08 s).

4.2.2. Spectral stability

If $d_{i,j}$ is the Euclidean distance between the MFCC vectors of frames i and j, the distance D_{stab}, defined as

$$D_{\text{stab}} = \min_{i \in \text{seg}} D_i, \quad \text{with} \quad D_i = \frac{d_{i,i-1} + d_{i,i+1}}{2} \tag{3}$$

is a measure of the maximum spectral stability observed inside the segment. The frame i_{s} where D_i is minimal is called the most stable frame of the segment. Our measurements reveal that FP segments have a smaller D_{stab} than NFP segments. The mean value of D_{stab} for FP segments is 5.18 (SD = 1.59), whereas it is 9.10 for NFPs (SD = 2.38).

4.2.3. Stable interval durations

Starting from i_{s}, the stable interval (SI) of a segment can be defined as the largest interval around i_{s} for which

$$d_{i,i_{\text{s}}} < T \quad \forall i \in \text{SI} \tag{4}$$

By selecting different values of T, one can determine different SIs and use the corresponding *stable interval durations* (SIDs) as segmental features. If $T \leqslant D_{\text{stab}}$ the SID is equal to zero. Filled pauses clearly tend to have a longer SID than other speech segments. In our experiments we have considered the SIDs for $T = 8$, 10, 12, 14, 16 and 18 as six distinct segmental features. The mean value of the SID for $T = 12$ is 3.19 frames for NFP (SD = 2.45) and 11.59 for FP (SD = 8.01).

4.2.4. Silence before and after the FP

Another prosodic cue that was found to be effective for the detection of FPs is the presence of a silence (sil) in an adjacent segment (either before or after the segment under test). A silence is defined as an interval during which the log-energy stays within 3 dB of the noise floor. Table 1 reveals that 80% of the FP segments are delimited by at least one silence, whereas only 61% of the NFP segments are. We found that the adjacent silences are also longer in the case of FP segments. It is even so that the post-FP silences are bound to be longer (mean of 0.19 s and SD of 0.23 s) than the pre-FP silences (mean of 0.13 s and SD of 0.16 s). For the NFP silences we found a mean of 0.11 s and a SD of 0.20 s.

Table 1
Number of filled pauses with and without adjacent silences

	Sil before	No sil before	Total
Sil after	946	768	1714
No sil after	870	657	1527
Total	1816	1425	3241

4.2.5. Spectral center of gravity

Another acoustic feature that was examined is the center of gravity of the mean log mel-power spectrum observed in the stable interval of the segment (defined for a given T):

$$g_{\text{S}} = \frac{\sum_{m=1}^{M} m\widetilde{S}(m)}{\sum_{m=1}^{M} \widetilde{S}(m)} \tag{5}$$

In this equation, $\widetilde{S}(m)$ represents the log signal power in the mth sub-band of an M channel mel-scale filter-bank ($M = 24$). We found that a center of gravity of 16 or more is a very good counter-indication for an FP. The mean center of gravity was 10.47 (SD = 2.65) for FP and 13.04 (SD = 3.98) for NFP, respectively.

4.2.6. Simple filled pause model output

Another feature is the logarithm of the output of a four mixture GMM that was trained on all the frames belonging to filled pause intervals. The model inputs are the 12 MFCCs. The mean log score was −16.9 for FP (SD = 1.4) and −18.8 for NFP (SD = 2.6), respectively.

4.2.7. Features related to the pitch

Goto et al. (1999) was successful in the detection of FPs in Japanese spontaneous utterances on the basis of features which represent frame-level changes of the fundamental frequency and the spectral envelope. On a set of 100 sentences, each containing at least one FP, the measured recall and precision rates were 84.9% and 91.5%, respectively. We also investigated the discrimination capabilities of some segmental pitch features for the detection of FPs in Dutch spontaneous speech. The investigated features were

- Pitch regression coefficient (PRC)
  We defined the PRC as the slope of the best fitting line through the non-zero pitches of the frames in a certain segment. No clear distinction was found between FP and NFP.
- Pitch modulation variance (PMV)
  The PMV is defined as the variance of the differences between the non-zero pitch values and the pitches predicted by the linear regression model. Again this feature did not offer a significant discrimination between FP and NFP.
- Relative pitch ratio (RPR)
  It is often supposed that filled pauses exhibit a low pitch compared to the surrounding speech segments. Therefore, the mean of the non-zero pitch values is computed for each segment and the ratio between this pitch and the mean pitch of the N preceding and N succeeding segments is defined as the relative pitch ratio of that segment. If the pitch of a segment is zero the RPR of that segment is undefined and assumed to be 1, if one of the contextual segments has a zero pitch it is excluded from the mean pitch computation of these segments. For $N = 7$ the RPR showed indeed a small ability to discriminate between FP and NFP segments. The mean RPR for FP segments was 0.96 while it was 1.00 for the other segments.

Apparently, for a non-tonal language like Dutch, the pitch features are not that powerful for the FP/NFP classification. Moreover, since the inclusion of a pitch extractor adds complexity to the acoustic front-end of the recognizer, we decided not to consider any of the pitch features for our FP detector.

4.3. Segment classification strategy

Since we aim to build a detector that can estimate the posterior probability of having an FP, given the acoustic observations, we propose to perform a classification of segments by means of an MLP (multilayer perceptron) with one hidden layer and one output which is, after proper training, supposed to provide exactly that probability.

A problem with the error back-propagation training of an MLP is that one often gets poor results when the prior probabilities of the classes are very different. Since only 1% of our segments have an FP label, we definitely are in that situation. Therefore, we first try to isolate a large number of NFP segments by means of Gaussian mixture models (GMMs) and then we train the MLP on the remaining segments. We have trained two GMMs to model the likelihoods $f(\boldsymbol{x}|\mathrm{FP})$ and $f(\boldsymbol{x}|\mathrm{NFP})$ with x representing the 13 features discussed above. The FP model consisted of 8 and the NFP model of 64 mixtures with diagonal covariance matrices. Since there is a good estimate of $P(\mathrm{FP})$, the likelihoods can be converted into *posterior* probabilities $P(\mathrm{FP}|\boldsymbol{x})$ and $P(\mathrm{NFP}|\boldsymbol{x})$, respectively.

Segments whose posterior probability ratio, denoted as PPR and defined as $P(\mathrm{FP}|\boldsymbol{x})/P(\mathrm{NFP}|\boldsymbol{x})$ exceeds some threshold t_{PPR} are considered as candidate FP segments and are supplied to the MLP classifier. The others are considered as NFP segments. The number of candidate FP segments can be controlled by modifying the threshold (see Table 2). Given that there were 3255 FP and 344,945 NFP segments in total, it is clear that with $t_{\mathrm{PPR}} = 10^{-6}$ we can retrieve about 89% of the FPs while eliminating 74% of the NFP segments.

In order to give the MLP some idea about the spectral envelope of the segment to classify, the MLP is supplied with 25 features: the 13 features that were used by the GMM, and 12 MFCCs characterizing the most stable frame of the segment.

Since the MLP is presumed to estimate the posterior probability $P(\mathrm{FP}|\boldsymbol{x})$, segments are classified as FP if the MLP output exceeds some posterior probability threshold t_{PP}. The latter is used to control the desired balance between the recall and the precision of the FP detector.

4.4. FP detection results

The trained GMM–MLP tandem is evaluated on the previously mentioned CGN test corpus which

Table 2
Number of segments passing through the GMM filter

t_{PPR}	# FP segments	# NFP segments
10^{-4}	2429	41,528
10^{-5}	2659	65,169
10^{-6}	2879	91,066

Table 3
Precision and recall of the FP classification of the GMM–MLP tandem

t_{PP}	Precision (%)	Recall (%)
0.05	44.4	91.1
0.15	59.8	83.6
0.20	65.1	81.6
0.25	68.5	78.9
0.30	72.9	74.5
0.40	77.5	65.7
0.60	83.5	50.7

contains 445 FPs, 440 of which are longer than 0.15 s.

We tested the performance using a GMM filter with a threshold $t_{\mathrm{PPR}} = 10^{-6}$ and comprising an MLP with 15 sigmoidal hidden units (and thus $15 \times 26 + 16 = 406$ free parameters). The results of our tests are listed in Table 3 as a function of t_{PP}. The data show that a precision of about 73% can be reached with a recall of 75%. Obviously, one can also operate the FP detector at a high precision (and a moderate recall) or at a high recall (and a moderate precision) if requested.

Attempts to improve the MLP classifier by performing some additional embedded training iterations on a larger database also including data for which no manually verified segmentation is available were not successful. A further increase in the number of hidden units did not result in any substantial improvement of the classification accuracy either.

5. Recognition experiments

In this section we discuss the recognition experiments we conducted with both the internally and externally informed disfluency handling approaches.

For the internally informed strategies, we report results for American English (Switchboard) and Dutch (CGN). For the externally informed strategies however, only results for Dutch are presented.[1]

Before discussing the recognition accuracies, we briefly describe the baseline speech recognition system that we built and the Switchboard and CGN recognition tasks on which we have evaluated our systems.

5.1. The baseline speech recognition system

The recognition system that we used is the ESAT speech recognizer (Duchateau et al., 1998; Demuynck et al., 2000). It performs a single pass time synchronous beam search and it comprises gender independent acoustic models. A global phonetic decision tree defines a large number of tied states that are used in cross-word context and position dependent phoneme models. Each tied state is modeled with a mixture of on average 220 tied gaussians from a large set of state-independent gaussians. No speaker adaptation is applied.

The language model (LM) is a trigram backoff language model which is retrieved from text material and/or orthographic transcripts of spontaneous speech. Good-Turing is used as the smoothing technique. We chose to consider FPs as integral elements of the language because Pakhomov (1999) obtained good results with this technique. He compared it to a baseline technique discarding all FPs and he found that keeping the FPs caused a reduction of the WER from 32.2% to 28.5% for spontaneous medical dictation.

5.2. The Switchboard task

For the Switchboard task the acoustic models are learned on 310 h of Switchboard-1 data. The global phonetic decision tree implies 8K tied states. Per state, the gaussians are selected from a set of 117K gaussians. The lexicon consists of all the 27K words (including FP) appearing in the Switchboard-1 transcripts (3M word tokens), the LM is estimated on the same data.

The Switchboard test set (see Section 2) covers 20 phone calls, 1718 sentences and about 20K word tokens.

The software available at *http://www.nist.gov/speech/tools* is used to compute the WER and to assess the statistical significance of measured performance differences. Our baseline system obtains a WER of 29.8%.

5.3. The CGN recognition task

For the CGN task the acoustic models are learned on 44 h of Flemish spontaneous data from CGN. The global phonetic decision tree defines 3500 tied states. Per state, the gaussians are selected from a set of 32K gaussians. The lexicon consists of the 40K most frequent words derived from Dutch

[1] Since the research was sponsored by the Flemish Authority, the emphasis had to be on Dutch. However, the experiments on Switchboard are helpful to demonstrate that our baseline system exhibits a state-of-the-art recognition performance.

newspaper material and supplemented with words that are needed to attain a full lexical coverage on the test set. This way, the results presented here are not influenced by the presence of out-of-vocabulary words. The lexicon contains a pronunciation model of the filled pause. This model represents a number of pronunciation variants (without probabilities) that were obtained manually, such as @, @m, @@, @@m, @mm, @@@, … The LM is trained on newspaper material extended with 3M word tokens of spontaneous speech transcripts from the CGN.

When evaluating this recognition system, all FPs appearing in the reference and in the recognized word strings are removed, meaning that the WERs only measure the number of errors related to regular words. This approach which we will also adopt in all our further recognition experiments was also applied in (Schramm et al., 2003).

The test set (see Section 2.1) comprises speech of 27 speakers and it contains 7041 regular words and 445 filled pauses. Hence, the FP rate is 445/7496 or 5.94% and thus significantly larger than the 2.7% which was measured on the CGN training corpus. The WER obtained with our baseline system on the test set is equal to 36.1%. This means that the CGN task is more difficult than the Switchboard task. This can be due to the larger diversity of the data, the larger mismatch between the LM and the spontaneous data, the smaller size of the acoustic model training database, etc.

5.4. Experimental results with internally informed strategies

In a first experiment on Switchboard we have compared our three proposed LM context manipulation models individually with the baseline system (never modify the context) and with the standard context manipulation (SCM) system (always modify the context) proposed in (Stolcke and Shriberg, 1996). The results of this experiment are summarized in Table 4. They confirm that the standard method does not offer any significant improvement. They also show that our word repetition model yields a small but statistically significant improvement (highlighted result) whereas the hesitation and the restart model unfortunately are ineffective.

Table 4
WERs for the baseline system and for systems using standard context manipulation models (SCM) and newly proposed context manipulation models, respectively

Disfluency	Model	Baseline (%)	Standard (%)	Proposed (%)
Repetition	Repetition	29.8	29.7	**29.6**
Filled pause	Hesitation	29.8	29.9	29.8
	Restart	29.8	29.8	29.9

A more detailed analysis shows that the proposed repetition model changes less than 5% of the recognized sentences, but mostly in the right sense. The low number of changed recognition outputs is not that surprising given the low WR-rate in spontaneous speech (around 1.4% of the words, as shown in Section 2).

In a second experiment we repeated the same tests on the CGN corpus. With the standard context manipulation technique being applied on FPs, a WER of 35.9% was obtained (see Table 5). Using the repetition and the hesitation model the WERs were 35.9% and 35.8%, respectively. Both of these WERs are statistically lower than the baseline WER of 36.1%, but the differences remain small due to the small WR-rate (below 1%) observed in the CGN data.

In general we can conclude that none of the tested context manipulation models can cause substantial gains in recognition accuracy. However, for both tasks the best performances were obtained with one of the proposed models. The small gains of the two models that are triggered by a filled pause are attributed to the fact that these models rely on the detection of a filled pause by the decoder. In the current system it is fairly easy to hypothesize such a filled pause and thus to generate a prediction context modification at too many places where there is no disfluency in the signal.

A better alternative would be to create a separate acoustic model for the filled pause as a word-level

Table 5
WERs for systems using externally informed FP handling methods

System	Disfluency handling	WER (%)
BS (baseline system)	None	36.1
BS + SCM	Internal strategy	35.9
BS + repetition		35.9
BS + hesitation		35.8
BS + drop	External strategy	34.5
BS + LMA		34.6
BS + LMA + drop		34.3
BS + SCM + drop	Combination	34.3
BS + SCM + LMA		34.6
BS + LMA + drop + hesitation		34.1

unit, as was done in (Schramm et al., 2003). We tried this on the Switchboard task but without any success.

5.5. *Experimental results with externally informed strategies*

As already discussed before, it is expected that frame dropping requires a detector with a high precision (do not throw away useful frames) whereas LM adaptation (LMA) may profit more from a detector with a high recall (make it applicable at all places where an FP is likely to occur). Therefore, we have investigated the performance of the two proposed strategies in combination with the same FP detector but working at different operating points in the (precision, recall) plane.

By applying frame dropping in combination with an FP detector with a high precision (83.5%) it was possible to reduce the WER from 36.1% to 34.5% (see Table 5) which is a statistically significant reduction. If the precision is lowered to 50%, the WER increases to 35%.

By applying LM adaptation in combination with an FP detector with a high recall (91.1%) it was possible to reduce the WER from 36.1% to 34.6% (see Table 5) which is again a statistically significant reduction. For LMA, the attained performance gain depends on the values of the control parameters C_1, C_2 and D discussed in Section 3.2. The best choices for C_1 and C_2 are 1 and 0, respectively. The value of D is not critical: as long as $D > 0.2$ s, the performance gain changes by less than 0.2%. The advantage of imposing a small maximum delay D is of course that it constrains the maximum time delay introduced in the search.

5.6. *Combining the different FP handling strategies*

In the former section it was demonstrated that frame dropping and LM adaptation are about equally effective when used in combination with our external FP detector when configured in a good operating point. In the following experiments we investigate whether the two externally informed strategies can complement each other, and whether they can also be combined effectively with the internally informed strategies we proposed.

5.6.1. *Combining frame dropping and LM adaptation*

Since frame dropping is most effective with a high precision FP detector and LM adaptation with a high recall FP detector, it seems logical to apply frame dropping when an FP segment with a high score (>0.5) was generated by the FP detector, and LM adaptation when an FP segment with a moderate score (between 0.05 and 0.5) was generated.

With the proposed combination of techniques we were able to further reduce the WER a little bit, to 34.3% (see Table 5). Another advantage of the combination is that its results are almost independent of the choice of the control parameters. The effect of D is negligible and $C_1 = C_2 = 0$ seems to be the best choice for the other two parameters.

5.6.2. *Adding standard context manipulation*

Although SCM alone does not seem to cause any significant improvement, we did investigate whether this is still true when it is used in combination with either one of the two externally informed methods for handling FPs. The results in Table 5 show that this is indeed the case.

5.6.3. *Adding the hesitation and the repetition model*

Since the proposed hesitation model did offer a small improvement of the recognition accuracy we have investigated whether this improvement is still present when the model is used in combination with frame dropping and LM adaptation. The answer is that it does further reduce the WER to 34.1% (see Table 5).

We also performed a test with a system incorporating all the techniques: frame dropping, LM adaptation, hesitation and word repetition context manipulation, but with this system, we obtained a WER of 34.2%, meaning that the repetition model is not effective on top of all the other methods.

6. Additional experiments and discussion

Although our externally informed search strategies result in a reduction of 2% absolute of the WER, this reduction is not as spectacular as we anticipated when we started our research. In this section we first describe a detailed error analysis we performed in order to investigate the main reasons for this. We also describe an experiment which was designed to find out how much larger the improvement could be if a better FP detector were available.

We concentrate on filled pauses here since for these disfluencies we found by far the most substantial improvements.

6.1. Detailed error analysis

In order to perform our error analysis we have selected a small corpus of 118 CGN test sentences (3737 words) containing at least one filled pause in their reference transcription. In total this small corpus comprised 250 FPs.

For the evaluation of our disfluency handling methods we first of all measured the over-all WER and the local WER which we defined as the WER observed in short windows starting at the reference word in front of the filled pause and ending at the word just after that filled pause. By comparing these two WERs, it should be possible to check whether or not FPs cause extra problems for the recognizer. In order to find out whether an FP is likely to trigger a chain reaction, we have also counted the number of consecutive regular word errors that were produced in the vicinity of each FP. The following example of an aligned reference and recognized sentence pair was found in our CGN test data:

reference:	…	elk	jaar	uh	of	tot	de	twee	…
recognized:	…	elk	jaar	*u*	*op*	*met*	de	*tee*	…

Translated to English, this gives

reference:	…	every	year	uh	or	to	the	two	…
recognized:	…	every	year	*you*	*on*	*with*	the	*tea*	…

If one first removes the FP from the reference sentence before analyzing the errors, it becomes clear that in the above example (Dutch sentences) there are two local errors (the insertion of "u" and the substitution of "op") and three consecutive word errors (the insertion of "u" and the substitutions of "of" and "tot").

The results for two systems (baseline and best system) are listed in Table 6. Apparently the over-all WERs obtained for the small subcorpus are very representative of the WERs obtained for the full corpus, but this is of course not the most important observation to retrieve from the table. The more important observations are the following:

Table 6
Detailed error analysis: the over-all WER, the local WER (in the vicinity of an FP) and the number of consecutive word errors per FP

System	Global WER	Local WER	Word errors/FP
BS	36.5	56.7	1.50
Best	34.3	42.0	1.17

(1) The local WER of the baseline system is substantially higher than its over-all WER. We consider this as a clear support of our hypothesis that FPs cause particular problems for the recognizer.
(2) The expected chain reaction is less pronounced than originally anticipated: the average number of errors induced by an FP is only 1.5, meaning that the maximum gain in performance attainable with FP handling strategies is bound to be smaller than 1.5 times the FP rate.
(3) Our best disfluency handling strategy does have a significant impact on the local WER (a reduction of 14.7% absolute), meaning that it does what it is supposed to do, namely deal with problems due to the presence of an FP.

We will now try to make a realistic estimate of the maximum performance gain that can be achieved on our test data by means of disfluency handling methods. If all the regular word errors occurring in the vicinity of an FP were effectively caused by the presence of that FP, the maximum gain would be 1.5 errors per FP.

However, if we randomly select 250 regular reference words and if we count the associated number of consecutive word errors in the same way as we did with the FPs, we find a number of 0.8 errors per word. Consequently, we argue that the number of errors that is on average induced by the presence of an FP is of the order of $1.5 - 0.8 = 0.7$.

Given that the FP rate in our data is 5.94%, this would finally lead to a maximum attainable gain in over-all WER of about 4.1% for these data. Note that this gain is about two times larger than the gain of 2% we actually obtain with our best system.

This last result contrasts a bit with the fact that the local WER of our best system is already pretty close to the over-all WER of the baseline system. Apparently our FP handling methods introduce a number of new errors in areas not corresponding to an FP in the speech. If we would be able to conceive a better external FP detector, we would expect less of these errors.

6.2. Impact of the external FP detector

In order to confirm the above hypothesis, we have evaluated the two externally informed FP handling methods in combination with a 'perfect' FP detector which we define as the FP detector generating the manually labeled FP segments (provided with the CGN data) with an FP score of 1. In Table 7 we have collected the recognition performances when the FP handling systems are being supplied with the real and the perfect FP detector outputs respectively. The improvements with respect to the baseline system are expressed in terms of the WER and the average number of corrected words per FP, denoted as #cwrds/FP.

Our results (Table 7) do confirm that with a perfect FP detector, the WER can be reduced by 4.1% (0.7 corrections per FP) which is equal to the estimated upper bound. They also demonstrate that this gain can only be achieved by means of frame dropping since LM adaptation alone never yields more than 0.3 corrected words per FP. We argue that the latter result demonstrates that the decoder part of the speech recognizer itself is not able to give a correct interpretation to the FP frames it is confronted with.

Table 7
Achievable gains of two externally informed methods using a real and a perfect FP detector, respectively

System	Perfect FP detector		Real FP detector	
	WER (%)	# cwrds/FP	WER (%)	# cwrds/FP
BS + drop	32.0	0.70	34.5	0.27
BS + LMA	34.4	0.30	34.6	0.25

6.3. Dependency of the FP rate

We also investigated whether there is a correlation between the attained performance gain and the FP rate of the speech utterances. Therefore we have performed a recognition experiment on 27 speakers selected from the training corpus on the basis of their FP rate. Note that these speakers were not involved in the training of the acoustic models nor the language model of the recognizer.

We have divided this set into five subsets by grouping the speakers on the basis of their FP rate. Table 8 gives an overview of the characteristics of these five databases and the WERs obtained with some of our systems on these databases. Apparently, the data are significantly more difficult to recognize than the test data we used before. The average performance gain due to our disfluency handling methods is also smaller (around 0.9% absolute) than before, but it is still statistically significant.

The bottom row of Table 8 shows that the improvement of the best system over the baseline system is roughly correlated with the FP rate. The figures also show that frame dropping starts to harm the performance if the FP rate gets too high: the best system for database 4 and 5 is the one without frame dropping whereas for database 1–3 it is the one incorporating both frame dropping and LM adaptation.

Table 8
Influence of the FP rate on the WERs obtained with some of the proposed methods

	db 1	db 2	db 3	db 4	db 5
FP rate (%)	1.97	4.16	5.57	7.57	8.85
# Words	5269	5711	5188	7089	4373
# Speakers	6	6	5	5	5
# FPs	106	248	306	581	425
WER (BS)	35.50	47.66	44.79	36.12	44.31
WER (BS + LMA)	35.29	47.30	43.75	34.78	43.45
WER (BS + LMA + hesitation)	35.25	47.28	43.77	**34.60**	**43.33**
WER (BS + LMA + drop + hesitation)	**34.83**	**47.22**	**43.38**	35.13	43.78
WER (BS) − WER (best)	0.7	0.4	0.4	1.5	1.0

6.4. Dependency on the baseline WER

In the course of our research we have tested FP handling methods in combination with baseline systems having a lower recognition accuracy (because they did not yet incorporate a spontaneous speech language model or because their acoustic models were trained on read speech only). From these tests it follows that the improvement induced by our methods did not decrease when better baseline systems became available. On the contrary, while the baseline WER could be reduced from 45.6% to 36.1%, the improvement induced by our methods actually increased from 1.7% to 2%. This is a hopeful result in view of the expectation that acoustic and linguistic models of spontaneous speech will further improve now that more and more spontaneous speech corpora are becoming available to the speech community.

7. Conclusion

In this paper we have proposed different strategies for coping with disfluencies in the search engine of a spontaneous speech recognizer. We made a distinction between internally informed approaches that totally rely on the standard knowledge sources (acoustic models, pronunciation models and language model) and externally informed approaches that also take into account evidence for disfluencies as it emerges from an external acoustic preprocessor.

We started by developing some new internally informed approaches which boil down to context manipulation methods, described at the level of the language model. New words can be predicted on the basis of a context with or without taking into account the disfluencies hypothesized in the recent past. The basic principle underlying the new methods is to provide alternative paths in the search network when a disfluency is hypothesized, and to leave it to the search engine to choose the best path. From experiments on Switchboard and on CGN it became clear that the proposed approaches only lead to a rather small (often not statistically significant) improvement of the recognition accuracy.

Since the recognizer may be insufficiently selective in hypothesizing the disfluencies, we developed a specialized preprocessor which operates independently of the search and which searches for FPs on the basis of acoustic and prosodic features that are not accessible to a standard recognizer. After having selected the appropriate features we could build an FP detector that is capable of detecting a large fraction of the filled pauses with a high precision. Then we proposed two strategies for incorporating the posterior probabilities at the output of this detector into the search engine.

Experiments on Flemish spontaneous speech showed that the externally informed approaches for handling FPs yield a moderate but statistically significant gain in recognition performance: the error rate could be reduced from 36.1% to 34.3%. This improvement corresponds to the correction, on average, of about 0.3 regular word errors per FP occurring in the speech. This is still quite below the maximum of 0.7 word corrections per FP we have been able to estimate on the basis of a statistical analysis of the error patterns we observed in the vicinity of FP intervals and regular word intervals.

In combination with the proposed internally informed methods for FP and WR handling, we could finally obtain a WER of 34.1%. A detailed analysis of our results has demonstrated that (1) the extend of the improvements is correlated with the disfluency rate of the speech, (2) the attained improvement does not decrease when the WER of the baseline system decreases, and last but not least, (3) the largest improvement obtained thus far is about 43% of the improvement that could be achieved with an ideal (manual) FP detector. The latter conclusion implies that significantly larger improvements are attainable if a significantly better FP detector could be conceived.

Acknowledgments

This research was supported by the Flemish Institute for the Promotion of Scientific and Technical Research in the Industry (ATraNoS project, contract ADV/STWW/000151). More information on the ATraNoS project can be found at http://www.esat.kuleuven.ac.be/psi/spraak/projects/ATraNoS).

References

Adda, G., Jardino, M., Gauvain, J., 1999. Language modeling for broadcast news transcription. In: Proc. European Conference on Speech Communication and Technology, Vol. IV, Budapest, Hungary, pp. 1759–1762.

Adda-Decker, M., Habert, B., Barras, C., Adda, G., Boula de Mareuil, P., Paroubek, P., 2003. A disfluency study for cleaning spontaneous speech automatic transcripts and improving speech language models. In: Proc. of ITRW on

Disfluency in Spontaneous Speech, Göteborg, Sweden, pp. 67–70.

Batliner, A., Kiessling, A., Burger, S., Nöth, E., 1995. Filled pauses in spontaneous speech. In: Proc. International Congress of Phonetic Sciences, Stockholm, Sweden.

Beyerlein, P., Aubert, X., Haeb-Umbach, R., Harris, M., Klakow, D., Wendemuth, A., Molau, S., Pitz, M., Sixtus, A., 1999. The Philips/RWTH system for transcription of broadcast news. In: Proc. European Conference on Speech Communication and Technology, Vol. II, Budapest, Hungary, pp. 647–650.

Demuynck, K., Duchateau, J., Van Compernolle, D., Wambacq, P., 2000. An efficient search space representation for large vocabulary continuous speech recognition. Speech Comm. 30 (1), 37–53.

Duchateau, J., Demuynck, K., Van Compernolle, D., 1998. Fast and accurate acoustic modelling with semi-continuous HMMs. Speech Comm. 24 (1), 5–17.

Duchateau, J., Laureys, T., Demuynck, K., Wambacq, P., 2003. Handling disfluencies in spontaneous language models. In: Gaustad, T. (Ed.), Computational Linguistics in The Netherlands. Language and Computers. Studies in Practical Linguistics. Rodopi, Amsterdam, The Netherlands and New York, USA, pp. 39–50.

Gabrea, M., O'Shaugnessy, D., 2000. Detection of filled pauses in spontaneous conversational speech. In: Proc. International Conference on Spoken Language Processing, Vol. III, Beijing, China, pp. 678–681.

Gauvain, J., Lamel, L., Adda, G., Jardino, M., 1999. Recent advances in transcribing television and radio broadcasts. In: Proc. European Conference on Speech Communication and Technology, Vol. II, Budapest, Hungary, pp. 655–658.

Godfrey, J., Holliman, E., McDaniel, J., 1992. Switchboard: Telephone speech corpus for research and development. In: Proc. International Conference on Acoustics, Speech and Signal Processing, Vol. I, San Francisco, USA, pp. 517–520.

Goedertier, W., Goddijn, S., Martens, J.-P., 2000. Orthographic transcription of the spoken Dutch corpus. In: International Conference on Language Resources and Evaluation, Athens, Greece, pp. 909–914.

Goto, M., Itou, K., Hayamizu, S., 1999. A real-time filled pause detection system for spontaneous speech. In: Proc. European Conference on Speech Communication and Technology, Vol. I, Budapest, Hungary, pp. 227–230.

Ma, K., Zavaliagkos, G., Meteer, M., 2000. Bi-modal sentence structure for language modeling. Speech Comm. 31 (1), 51–67.

Martens, J.-P., Binnenpoorte, D., Demuynck, K., Van Parys, R., Laureys, T., Goedertier, W., Duchateau, J., 2002. Word segmentation in the spoken Dutch corpus. In: International Conference on Language Resources and Evaluation, Vol. V, Las Palmas, Canary Islands, Spain, pp. 1432–1437.

O'Shaugnessy, D., 1993. Locating disfluencies in spontaneous speech: an acoustical analysis. In: Proc. European Conference on Speech Communication and Technology, Vol. III, Berlin, Germany, pp. 2187–2190.

Pakhomov, S.V., 1999. Modeling filled pauses in medical dictations. In: Proc. Association for Computational Linguistics (ACL), College Park, Maryland, USA, pp. 619–624.

Peters, J., May 2003. Lm studies on filled pauses in spontaneous medical dictation. In: Proc. Human Language Technology conference/North American Chapter of the Association for Computational Linguistics Annual Meeting, Edmonton, Canada, pp. 82–84.

Quimbo, F.C., Kawahara, T., Doshita, S., 1998. Prosodic analysis of fillers and self-repair in Japanese speech. In: Proc. International Conference on Spoken Language Processing, Sydney, Australia, pp. 3313–3316.

Schramm, H., Aubert, X.L., Meyer, C., Peters, J., 2003. Filled-pause modeling for medical transcriptions. In: Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition, Tokyo, Japan.

Shriberg, E., 1996. Disfluencies in Switchboard. In: Proc. International Conference on Spoken Language Processing, Vol. Addendum, Philadelphia, USA, pp. 11–14.

Shriberg, E., Stolcke, A., 1996. Word predictability after hesitations: a corpus-based study. In: Proc. International Conference on Spoken Language Processing, Vol. III. Philadelphia, USA, pp. 1868–1871.

Siu, M., Ostendorf, M., 1996. Modeling disfluencies in conversational speech. In: Proc. International Conference on Spoken Language Processing, Vol. I, Atlanta, USA, pp. 386–389.

Stolcke, A., Shriberg, E., 1996. Statistical language modeling for speech disfluencies. In: Proc. International Conference on Acoustics, Speech and Signal Processing, Vol. I, Atlanta, USA, pp. 405–408.

Stouten, F., Martens, J.-P., 2003. A feature-based filled pause detection system for Dutch. In: Proc. IEEE Automatic Speech Recognition and Understanding Workshop, Virgen Islands, USA, pp. 309–314.

Stouten, V., Van hamme, H., Duchateau, J., Wambacq, P., 2003. Evaluation of model-based feature enhancement on the AURORA-4 task. In: Proc. European Conference on Speech Communication and Technology, Geneva, Switzerland, pp. 349–352.

Vorstermans, A., Martens, J.-P., Van Coile, B., 1996. Automatic segmentation and labeling of multi-lingual speech data. Speech Comm. 19, 271–293.

Yu, H., Tomokiyo, T., Wang, Z., Waibel, A., 2000. New developments in automatic meeting transcription. In: Proc. International Conference on Spoken Language Processing, Vol. IV, Beijing, China, pp. 310–313.

Zechner, K., Waibel, A., 1998. Using chunk based partial parsing of spontaneous speech in unrestricted domains for reducing word error rate in speech recognition. In: Proc. 17th Conference on Computational Linguistics (COLING/ACL'98), Montreal, Canada, pp. 1453–1459.