

# Early error detection on word level

Gabriel Skantze & Jens Edlund

CTT, KTH, Sweden

{gabriel,edlund}@speech.kth.se

## Abstract

In this paper two studies are presented in which the detection of speech recognition errors on the word level was examined. In the first study, memory-based and transformation-based machine learning was used for the task, using confidence, lexical, contextual and discourse features. In the second study, we investigated which factors humans benefit from when detecting errors. Information from the speech recogniser (i.e. word confidence scores and 5-best lists) and contextual information were the factors investigated. The results show that word confidence scores are useful and that lexical and contextual (both from the utterance and from the discourse) features further improve performance.

## 1. Introduction

In order to build robust spoken dialogue systems (SDSs), techniques must be developed to prevent, detect, and recover from errors. In large vocabulary conversational SDSs, automatic speech recognition (ASR) is an inevitable source of errors, and detecting ASR errors is an important part of error handling. The detection of errors can prevent over-generation of interpretations and thereby misunderstanding of utterances.

The term error detection has been used to denote different things. In what we will call *early error detection*, the task is to detect errors in the current result of some process – in this case, the ASR. Information from such detection can be used to select grounding strategy [1] and to signal understanding or, if the result is too corrupted, signal non-understanding. Another meaning of the term is what we call *late error detection*, in which case the system has already signalled understanding and the user has acted upon this signal. Based on the user’s reaction, the task is to detect whether the previous interpretation was incorrect, in which case a misunderstanding has occurred. Detection of such errors will probably initiate an error recovery process. The term error detection has also been used for *prediction* of errors that will occur at a later stage in the dialogue so that preventive actions may be taken.

This paper concerns early error detection – specifically the detection of insertions in the ASR output. It is important to note that this does not include error correction (i.e. replacing substitutions and re-inserting deletions). The task is just to determine if a given recognised word was present in the original utterance or not.

An obvious argument against early error detection as a post-processing step on the ASR output is that the problems that these techniques attempt to fix should be addressed in the ASR. However, as argued in [2], post-processing may consider systematic errors in the language and acoustic models, which arise from mismatched training and usage conditions. It is not always easy to find and correct the actual problems in the models and a post-processing algorithm may help to pin-

point them. Post-processing may also include factors that were not considered by the speech recogniser, such as prosody, semantics and dialogue history.

The most commonly used feature for early error detection is the ASR confidence score. A potential approach to early error detection using more features is to use a machine learning (ML) algorithm. ML has been used for both early [3], [4] and late [5] error detection, as well as error prediction [6]. These studies have all focussed on full utterances. More precisely, the task has been to decide whether the word error rate (WER) and/or concept error rate is greater than zero. This is useful for systems where short utterances are expected and their complexity limited. However, when long and complex utterances are expected and an n-gram language model is used for the ASR, most utterances can be expected to contain some errors. Long utterances may also contain more than one concept, rendering an all-or-nothing distinction too blunt. If some content words are intact, the recognition may still prove useful. The question is, then, which words are correct and which are not. In this paper, two machine learning algorithms have been used for early error detection on word level: transformation-based [7] and memory-based [8] learning.

A main issue for ML is which factors the learning can and should be based on, and how to operationalise these factors as features. Some features, such as dialogue history, may seem useful for error detection, but are hard to operationalise, especially for longer contexts. Finding whether a factor contribute to the performance of a human subject doing the error detection task may provide some guidance as to its value to the ML task. In the second study, we investigated which factors humans benefit from when detecting errors. Information from the ASR (i.e. word confidence scores and 5-best lists) and contextual information were the factors investigated.

## 2. Corpus collection

The studies presented are based on a corpus collected in an experiment on human error handling strategies [9], which is presented here briefly for completeness. The experiment was a modified Wizard of Oz set-up for testing error handling in dialogues where the operator (i.e. the wizard) gave Map-Task-like [10] directions to the user. The task for the user was to navigate through a simulated campus. The operator had access to a map over the campus and was given the task of guiding the user.

In order to elicit error-handling strategies, the user spoke through an off-the-shelf speech recogniser, and the operator could read the recognition results, but not hear the utterances. The operators had no experience in designing dialogue systems and did not have an understanding of how errors are traditionally handled in SDSs. This was deliberate, as the purpose of the study was to investigate human error handling strategies.

For the same reason, the operators were allowed to speak freely and the users were openly informed that they interacted

with a human operator, unlike the traditional Wizard of Oz setting. They were also aware that the ASR had limitations. The operators' speech was distorted by a vocoder. The user and operator were not allowed to see each other before the experiment, in order to minimise common ground.

16 subjects grouped in eight pairs were used. Each pair did five sessions, resulting in 40 dialogues. The average dialogue contained about 18 user utterances, with a mean utterance length of 6.8 words. The WER was fairly high, about 42%, due to the users' unrestricted speech and the fact that the bigram language model used was limited.

### 3. Study I: Machine learning

In this study, transformation-based and memory-based learning were used for early error detection. In transformation-based learning, the algorithm learns a set of transformation rules that are applied after each other [7]. All instances are initially tagged with the most common class. During training, the algorithm instantiate rule templates and creates a cascade of rules that most efficiently transforms the classes in the material in a positive direction. In the current study,  $\mu$ -TBL [7] was used for transformation-based learning.

In memory-based learning (also called instance-based learning), the training set is just stored as examples for later evaluation [8]. The instance to be classified is then compared to all examples to find the (set of) nearest neighbour(s). To measure the distance between two instances, the vectors of features for the instances are compared. In this study, TiMBL [8] was used for memory-based learning.

The classification task was to determine whether a given recognised word was present at the corresponding location in the spoken utterance (TRUE) or not (FALSE). For this study, the recognition results from the corpus were aligned to the transcription (using minimum edit distance) in order to determine for each word if it was correct or not. 73.2% of the words turned out to be correct, which gives us a baseline to compare the machine learning performance with (this is the score that we would get if we tagged all words as TRUE). Of the 4470 words, 4/5 were used as training data and 1/5 as test data.

In Table 1, the features that were used for each word are classified into four groups: confidence, lexical, contextual and discourse. For dialogue act tagging, a simple set was constructed specifically for the domain. The content/non-content split was also made with the domain in mind. Content words were mainly nouns, adjectives and some verbs.

#### 3.1. Results

In order to investigate how the performance varied dependent on which features that were used, different combinations of feature set groups were used. The results are shown in Table 2. TiMBL seemed to perform best with the IB1 algorithm, gain ratio weighting and overlap as distance metric (except for confidence, for which a numeric distance metric was used). Depending on feature set, different values for k were best. Since  $\mu$ -TBL cannot automatically find thresholds for numeric values, a set of ten (equally sized) intervals were defined for the confidence score.

As the table shows, each group seems to add (more or less) to the performance.  $\mu$ -TBL seems to perform a bit better (although the difference has not been tested for significance).

Table 1: Features used for error detection.

Group	Feature	Explanation
Confidence	CONFIDENCE	Speech recognition word confidence score
Lexical	WORD	The word
	POS	The part-of-speech for the word
	LENGTH	The number of syllables in the word
	CONTENT	Is it a content word?
Contextual	PREVPOS	The part-of-speech for the previous word
	NEXTPOS	The part-of-speech for the next word
	PREVWORD	The previous word
Discourse	PREVDIALOGUEACT	The dialogue act of the previous operator utterance
	MENTIONED	Is it a content word that has been mentioned previously by the operator in the discourse?

Table 2: Performance of the machine learning algorithms depending on feature set.

Feature set	$\mu$ -TBL	TiMBL
Confidence	77.3%	76.0% (k=5)
Lexical	77.5%	78.0% (k=1)
Lexical + Contextual	81.4%	82.8% (k=1)
Lexical + Confidence	81.3%	81.0% (k=5)
Lexical + Confidence + Contextual	83.9%	83.2% (k=1)
Lexical + Confidence + Contextual + Discourse	85.1%	84.1% (k=1)

Table 3: The top rules learned by  $\mu$ -TBL.

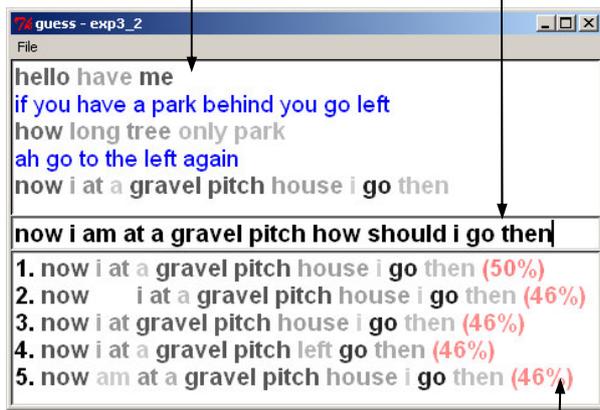
Transformation	Rule
TRUE > FALSE	CONFIDENCE < 50 & CONTENT = TRUE
TRUE > FALSE	CONFIDENCE < 60 & POS = Verb & LENGTH = 2
TRUE > FALSE	CONFIDENCE < 40 & POS = Adverb & LENGTH = 1
TRUE > FALSE	CONFIDENCE < 50 & POS = Adverb & LENGTH = 2
TRUE > FALSE	CONFIDENCE < 40 & POS = Verb & LENGTH = 1
FALSE > TRUE	CONFIDENCE > 40 & MENTIONED = TRUE & POS = Noun & LENGTH = 2

With the richest feature set,  $\mu$ -TBL performs 11.9% better than baseline.

The performance of the two machine learners seem to be very similar. In order to investigate whether they made the same mistakes, the result of the classifications were compared. In 69 cases, both learners made the same mistake, in 137 cases they disagreed. Thus, if a perfect ensemble method would be used that could choose the right classifier, the resulting performance would be 92.3%.

Since many interpretation modules in dialogue systems are mainly dependent on content words, the performance of these are important for detection. There were 285 content words in the test material of which 199 were correctly recog-

The dialogue so far. User utterances in greyscale and operator utterances in black. Correction field for the judge.



N-best list from the ASR. Utterance confidence score in parenthesis.

Figure 1: The judges' interface

nised. This gives a baseline of 69.8%. For these words, the best scores for the classifiers were 87.7% ( $\mu$ -TBL) and 87.0% (TiMBL). Thus, the best classifier  $\mu$ -TBL performs 17.9% better than baseline for content words. (A perfect ensemble method would score 94.4%.)

The top rules that were learned by  $\mu$ -TBL are shown in Table 3. The first rule learned states that all content words with confidence less than 50 should be tagged as false. The rest of the rules mainly concern different confidence thresholds depending on type of word (often represented with part-of-speech and word length). There are also some interesting discourse rules, such as the sixth: all two-syllables content nouns with a confidence score high enough that have been mentioned previously by the operator are (probably) correct.

#### 4. Study II: Human error detection

The features used in the machine learning study was chosen because they could intuitively contribute to error detection and they were easy to operationalise. However, it should be interesting to examine what features humans could benefit from in performing the task, especially features that are hard to operationalise. In this study, an experiment was conducted where human subjects (henceforth referred to as judges) were asked to detect and errors in ASR results. In order to investigate whether dialogue context, ASR confidence measures, and ASR N-best lists provide help when detecting errors, the judges' access to these factors was varied systematically.

##### 4.1. Method

Four dialogues with higher average WER than the corpus as a whole were chosen. The first 15 exchanges of these dialogues were used for the experiment, resulting in a subset of the corpus containing 60 exchanges. 50% of the words in the subset were correctly recognised, which gives the baseline for the task, by either deleting all words or leaving the entire string unaltered.

Eight judges with some limited experience in speech technology were asked to delete words in the ASR output that

Table 4: Context levels

NOCONTEXT	No context. ASR output only, utterances in random order.
PREVIOUSCONTEXT	Previous utterance from the operator visible. Utterance pairs in random order.
FULLCONTEXT	Full dialogue. The operator utterances and the ASR output are given incrementally and stay visible throughout the dialogue.
MAPCONTEXT	As FULLCONTEXT, with the addition of the map that was used by the interlocutors.

Table 5: ASR information levels

NOCONFIDENCE	Recognised string only.
CONFIDENCE	Recognised string, colour coded for word confidence (grey scale: dark for high confidence, light for low)
NBESTLIST	As CONFIDENCE, but the 5-best ASR result was given.

they believed to be wrong, using a custom-made tool. Figure 1 shows the tool in English translation.

Each judge assessed all four dialogues, with a different amount of visible context for each dialogue. The four levels of context are shown in Table 4.

Furthermore, each ASR result was repeated three times with an increasing degree of information from the ASR attached, and the judge had to reassess the recognition each time. The ASR information levels are listed in order of appearance in Table 5. The order of the dialogues and context levels were systematically varied for each judge.

The data consists of three versions of each recognised utterance: the reference (R), the recogniser hypothesis (H), and the judged hypothesis (J). To measure the judges' performance, J was aligned to R. The number of misrecognised words altered by the judge and correctly recognised words that were left unaltered were summed up for each utterance. This number was divided by the total number of words in the ASR result to yield a number between 0 and 1. 1 indicates that all incorrectly recognised words (insertions and substitutions) were detected and no correctly recognised words were judged as errors. 0 indicates the opposite: all correctly recognised words were judged as errors and all errors were judged correct.

##### 4.2. Results

There were main effects of both ASR information level and context level (two-way repeated measures ANOVA;  $p < 0.05$ ). Post tests revealed that NBESTLIST was better than CONFIDENCE, which in turn was better than NOCONFIDENCE. PREVIOUSCONTEXT was better than NOCONTEXT ( $p < 0.05$ ), but there was no difference between PREVIOUSCONTEXT, FULLCONTEXT and MAPCONTEXT. There were no interaction effects between variables. Overall, the judges performed significantly better than the baseline of 50%. The left column of Figure 2 shows mean error detection scores for the different ASR information and context levels. PREVIOUSCONTEXT, FULLCONTEXT and MAPCONTEXT were combined into CONTEXT, since they held no significant differences. To see what effect average WER had on the judges' results, the figures were recalculated over two subsets of the corpus: one

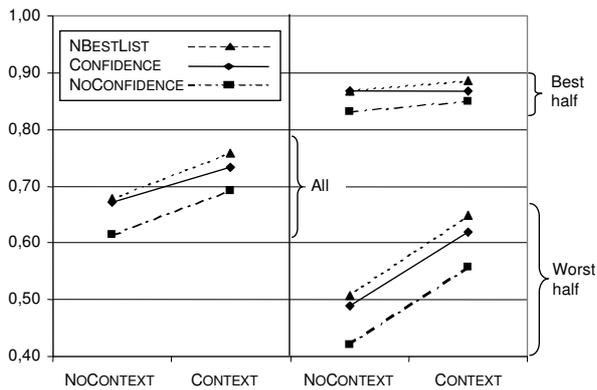


Figure 2: Mean error detection scores

containing the 30 utterances with the highest WER and another the 30 utterances with the lowest WER. Detection scores for the subsets are shown in the right column of Figure 2.

The effects for the worst utterances were the same as the effects in general. For the best utterances, the differences between different recognition information levels persisted. However, there were no significant differences between different context levels.

## 5. Conclusions & Discussion

Both studies show that word confidence scores are useful for early error detection, but that other features can be used to improve performance. Utterance context and lexical information improve the ML performance. The errors that are found with these features probably reflect errors in the language and acoustic models and should be corrected there, if possible. This is not always an easy task, however. Apart from using these methods for improving the performance of a specific application without collecting more data for models, a rule-learning algorithm such as  $\mu$ -TBL can be used to pinpoint the specific problems. For example, if the algorithm finds that a number of specific words should be classed as incorrect, these may be over-represented in the training material.

N-best lists are useful for human subjects. The question is how they should be operationalised to be included in the ML feature set. The discourse context of the utterance is potentially the most interesting feature, since it is not considered by the ASR. The ML improved only slightly from the discourse context, but the results from the second study suggests that the immediate discourse context of the utterance (i.e. the previous utterance) is the most important to humans for detection. For good recognitions, there was no effect from the discourse context, which indicates that the intact parts of a good recognition may provide sufficient context in themselves. For poorer recognitions, it seems that there is sufficient information in the previous utterance together with the judges' knowledge about the domain, and that further context is redundant. Thus, further work in operationalising context for ML should focus on the previous utterance. It could be argued that even though a long dialogue context does not improve the performance of humans, a machine may still be able to use it. Humans, however, generally seem to outperform machines when it comes to utilising context in spoken language.

In order to test if the performance of the ML is useful, the classifier should be tested together with a parser or keyword spotter to see if it can improve performance.

There are of course other features that should be investigated, such as prosody (speaking rate, loudness). These may improve performance further.

Since the classifiers disagree in so many cases, it would be interesting to test whether it would be possible to use an ensemble method that could pick the right classifier. This has been shown to be useful for other machine-learning tasks [11].

## 6. Acknowledgements

This research was carried out as part of the HIGGINS project (<http://www.speech.kth.se/higgins/>) at the Centre for Speech Technology, a competence centre at KTH, supported by VINNOVA (The Swedish Agency for Innovation Systems), KTH and participating Swedish companies and organizations.

## 7. References

- [1] Clark, H. H. & Schaefer, E. F. (1989). Contributing to Discourse. *Cognitive Science*, 13: 259-294.
- [2] Ringer, E. K., & Allen, J. F. (1997). Robust Error Correction of Continuous Speech Recognition. In *Proceedings of the ESCA-NATO Robust Workshop*.
- [3] Litman, D. J., Hirschberg, J., & Swertz, M. (2000). Predicting Automatic Speech Recognition Performance Using Prosodic Cues. In *Proceedings of NAACL*, 218-225.
- [4] Walker, M. A., Wright, J., Langkilde, I. (2000). Using Natural Language Processing and Discourse Features to Identify Understanding Errors in a Spoken Dialogue System. In *Proceedings of the 17th International Conference on Machine Learning*.
- [5] Krahmer, E., Swerts, M., Theune, T. & Weegels, M. E. (2001). Error Detection in Spoken Human-Machine Interaction. *International Journal of Speech Technology*, 4(1): 19-30.
- [6] Walker, M. A., Langkilde-Geary, I., Wright Hastie, H., Wright, J., & Gorin, A. (2002). Automatically Training a Problematic Dialogue Predictor for a Spoken Dialogue System. *Journal of Artificial Intelligence Research*, 16: 293-319.
- [7] Lager, T. (1999). The  $\mu$ -TBL System: Logic Programming Tools for Transformation-Based Learning. In *Proceedings of CoNLL-99*.
- [8] Daelemans, W., Zavrel, J., van der Sloot, K., & van den Bosch, A. (2003). *TiMBL: Tilburg Memory Based Learner, version 5.0, Reference Guide*. ILK Technical Report 03-10.
- [9] Skantze, G. (2003). Exploring Human Error Handling Strategies: Implications for Spoken Dialogue Systems. In *Proceedings of the ISCA Workshop on Error Handling in Spoken Dialogue Systems*.
- [10] Anderson, A. H., M. Bader, E. G. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. Thompson, A. R. Weinert (1991). The HCRC Map Task Corpus. *Language and Speech*, 34(4): 351-366.
- [11] Megyesi, B. (2002). *Data-driven syntactic analysis: methods and applications for Swedish*. PhD Thesis, KTH, Sweden.